

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN GIỮA KÌ**  
**NHẬP MÔN HỌC MÁY**

*Người hướng dẫn:* **TS. TRẦN LƯƠNG QUỐC ĐẠI**

*Người thực hiện:* **CHUNG THÁI KIỆT – 52200140**

**LÊ HÂN - 52200155**

**HUỲNH THANH BẢO NGỌC - 52200153**

**Lớp: 22050301**

**Khoá: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN GIỮA KÌ**  
**NHẬP MÔN HỌC MÁY**

*Người hướng dẫn:* **TS. TRẦN LƯƠNG QUỐC ĐẠI**

*Người thực hiện:* **CHUNG THÁI KIỆT – 52200140**

**LÊ HÂN - 52200155**

**HUỲNH THANH BẢO NGỌC - 52200153**

**Lớp: 22050301**

**Khoá: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

## LỜI CẢM ƠN

Trước hết chúng em xin gửi lời cảm ơn đến thầy *Trần Lương Quốc Đại*, giảng viên phụ trách bộ môn *Nhập môn Học Máy* em chân thành cảm ơn thầy đã luôn tận tình, nhiệt huyết giúp đỡ sinh viên chúng em trong môn học này. Trong quá trình học thầy luôn cố gắng giảng dạy cẩn thận và luôn hỏi mọi người đã hiểu hết chưa trước khi qua chương mới để đảm bảo chúng em vững kiến thức khi tiếp tục qua bài học khác. Và thầy cũng cung cấp thêm cho tụi em những kiến thức, hiểu biết đủ để thực hiện bài báo cáo cuối kì này. Nhóm chúng em sẽ cố gắng hết sức để thực hiện bài báo cáo chính chu nhất, tuy nhiên nếu có nhiều thiếu sót chúng em cũng rất mong được sự đóng góp ý kiến của thầy để bài báo cáo của nhóm em được hoàn thiện hơn.

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS. *Trần Lương Quốc Đại*. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Chung Thái Kiệt*

*Lê Hân*

*Huỳnh Thanh Bảo Ngọc*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(kí và ghi họ tên)

## MỤC LỤC

LỜI CẢM ƠN .....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	iii
MỤC LỤC .....	1
CHƯƠNG 1 - CÂU 1 .....	5
1.1    Giới thiệu vấn đề.....	5
1.1.1    Giới thiệu dataset.....	5
1.1.2    Giới thiệu bài toán .....	6
1.2    Phân tích dữ liệu .....	7
1.2.1    Khai phá và trực quan hóa dữ liệu .....	7
1.2.2    Tiền xử lý dữ liệu .....	21
1.3    Giới thiệu mô hình và tham số .....	24
1.3.1    Giải thích mô hình .....	24
1.3.2    Giải thích metric .....	30
1.4    Đánh giá và so sánh kết quả .....	36
1.4.1    Phân chia dữ liệu .....	36
1.4.2    Hồi quy .....	36
1.4.3    Phân loại .....	48
CHƯƠNG 2 - CÂU 2 .....	56
2.1    Tổng quan overfitting.....	56
2.1.1    Khái niệm overfitting .....	56
2.1.2    Bias và variance .....	56
2.1.3    Lý do overfitting .....	57
2.1.4    Biện pháp.....	59

2.2	Sử dụng hyperparameter tuning, cross-validation, regularization vào model .....	60
2.2.1	Hồi quy .....	60
2.2.2	Phân loại .....	67
2.2.3	Sử dụng model mới cho phân loại .....	69
CHƯƠNG 3 – CÂU 3.....		71
3.1	Giới thiệu feature selection using correlation analysis .....	71
3.1.1	Feature selection là gì? .....	71
3.1.2	Feature selection using correlation analysis là gì? .....	72
3.1.3	Áp dụng vào bài toán .....	74
TÀI LIỆU THAM KHẢO .....		77

## DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

### DANH MỤC HÌNH

Hình 1-1 Mô tả thông tin dataset .....	8
Hình 1-2 Trước và sau xử lý <code>pets_allowed</code> .....	9
Hình 1-3 Trước và sau xử lý <code>cityname</code> và <code>state</code> .....	9
Hình 1-4 Trước và sau khi xử lý <code>amenities</code> .....	10
Hình 1-5 Các giá trị NaN của dữ liệu sau khi xử lý.....	11
Hình 1-6 Mô tả giá trị numerical của <code>bathrooms</code> , <code>bedrooms</code> , <code>price</code> , <code>square_feet</code> .....	12
Hình 1-7 Tần suất của các giá trị numerical .....	13
Hình 1-8 Phân phối của <code>price</code> .....	14
Hình 1-9 Mô tả số lượng phần tử của các thuộc tính .....	19
Hình 1-10 Mô tả tổng quan phân bố giá nhà .....	20
Hình 1-11 Mô tả chi tiết phân bố giá nhà.....	21
Hình 1-12 Mô tả kết quả One-Hot Encoding.....	22
Hình 1-13 Mô tả Overfitting và Underfitting .....	31
Hình 1-14 Ma trận nhầm lẫn.....	35
Hình 1-15 So sánh MAE, RMSE, <b>R<sup>2</sup></b> .....	36
Hình 1-16 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Linear Regression.....	38
Hình 1-17 Learning Curve của Linear Regression .....	39
Hình 1-18 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Random Forest .....	41
Hình 1-19 Learning Curve của Random Forest.....	42
Hình 1-20 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Decision Tree:.....	44
Hình 1-21 Learning Curve của Decision Tree.....	45



Hình 1-22 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Gradient Boosting.....	46
Hình 1-23 Learning Curve của Gradient Boosting .....	47
Hình 1-24 Confusion Matrix.....	48
Hình 1-25 Learning Curve của Logistic Regression.....	51
Hình 1-26 Learning Curve của Decision Tree.....	52
Hình 1-27 Learning Curve của K-Nearest Neighbors (KNN).....	54
Hình 2-1 Mô tả bias và variance .....	57
Hình 2-2 Các tham số trước và sau xử lý của Random Forest Regressor.....	61
Hình 2-3 Learning Curve trước và sau điều chỉnh của Random Forest Regressor .....	61
Hình 2-4 Các tham số trước và sau xử lý của Decision Tree Regressor.....	63
Hình 2-5 Learning Curve trước và sau điều chỉnh của Decision Tree Regressor .....	63
Hình 2-6 Các tham số trước và sau xử lý của Gradient Boosting Regressor .....	65
Hình 2-7 Learning Curve trước và sau điều chỉnh của Gradient Boosting Regressor ..	65
Hình 2-8 Learning Curve trước và sau điều chỉnh của Decision Tree Classifier.....	67
Hình 2-9 Learning Curve trước và sau điều chỉnh của XGBClassifier .....	69
Hình 2-10 Accuracy .....	70
Hình 3-1 Tương quan các thuộc tính trong dataset.....	74
Hình 3-2 Tương quan của các thuộc tính lựa chọn .....	75
Hình 3-3 $R^2$ và MAE trước và sau chọn thuộc tính .....	75
Hình 3-4 Learning Curve trước và sau điều chỉnh.....	76

# CHƯƠNG 1 - CÂU 1

## 1.1 Giới thiệu vấn đề

### 1.1.1 Giới thiệu dataset

Dataset “Apartment for Rent Classified” từ UCI Machine Learning Repository cung cấp thông tin về các căn hộ cho thuê, được trích từ các trang web, nền tảng bất động sản của Mỹ. Bộ dữ liệu chứa 10000 hàng và 22 cột, dữ liệu gồm các thuộc tính sau:

- Id: integer
- Category: categorical
- Title: categorical
- Body: categorical
- Amenities: categorical
- Bathrooms: float
- Bedrooms: float
- Currency: categorical
- Fee: categorical
- Has\_photo: categorical
- Pet\_allowed: categorical
- Price: integer
- Price\_type: categorical
- Price\_display: categorical
- Square\_feet: integer
- Address: categorical
- Cityname: categorical
- Latitude: float
- Longitude: float

- Source: categorical
- Time: integer

### 1.1.2 Giới thiệu bài toán

#### 1.1.2.1 Bài toán hồi quy:

**Mục tiêu:** Dự đoán diện tích căn hộ dựa trên các đặc trưng như giá, số phòng tắm, phòng ngủ, tiện ích và vị trí địa lí...

**Mô tả:** Với dữ liệu này, mô hình hồi quy sẽ học cách ước tính diện tích từ các thuộc tính đầu vào. Các phương pháp hồi quy phổ biến như hồi quy tuyến tính, hoặc các mô hình phi tuyến như cây quyết định và hồi quy ngẫu nhiên, Gradient Boosting có thể được áp dụng.

**Ứng dụng:** Tìm kiếm và ra quyết định khi mua bất động sản. Người dùng cung cấp ngân sách, vị trí, số phòng ...Sau đó hệ thống sẽ trả về diện tích ước tính của căn hộ.

#### 1.1.2.2 Bài toán phân loại:

**Mục tiêu:** Phân loại tiện ích (amenities) của một căn hộ dựa trên đặc trưng như giá, số phòng tắm, phòng ngủ, diện tích vị trí địa lí.

**Mô tả:** Bài toán phân loại này nhằm phân chia các căn hộ vào 2 nhóm

- basic: các tiện ích cơ bản (chẳng hạn như không có hồ bơi, phòng gym)
- luxury: các tiện ích cao cấp (có hồ bơi, sân tennis...)
- Các phương pháp phân loại phổ biến như hồi quy logistic, cây quyết định, hoặc các mô hình phức tạp hơn như SVM hoặc rừng ngẫu nhiên có thể được áp dụng.

**Ứng dụng:** Tìm kiếm và ra quyết định khi mua bất động sản. Người dùng cung cấp ngân sách, vị trí, số phòng... Sau đó hệ thống sẽ xuất ra căn hộ loại tiện ích phù hợp.

## 1.2 Phân tích dữ liệu

### 1.2.1 Khai phá và trực quan hóa dữ liệu

#### 1.2.1.1 Lý thuyết

**Khai phá và trực quan hóa dữ liệu** là hai bước quan trọng trong quá trình phân tích dữ liệu, giúp các nhà phân tích hiểu sâu hơn về bản chất và đặc điểm của dữ liệu.

- **Khai phá dữ liệu** cho phép chúng ta khám phá các thuộc tính, kiểm tra phân bố, phát hiện mối quan hệ giữa các biến và xác định những giá trị bất thường. Thông qua phân tích thống kê mô tả và phân tích tương quan, chúng ta có thể hiểu rõ các yếu tố trong tập dữ liệu và đưa ra các kết luận sơ bộ.
- **Trực quan hóa dữ liệu** đóng vai trò như một công cụ truyền tải, biến những con số phức tạp thành các hình ảnh trực quan như biểu đồ phân tán, biểu đồ hộp và biểu đồ nhiệt. Các biểu đồ này không chỉ giúp dễ dàng nhận diện các xu hướng và mẫu trong dữ liệu mà còn giúp các bên liên quan, từ nhà nghiên cứu đến nhà quản lý, có thể nhanh chóng hiểu và đưa ra quyết định.

**Khai phá và trực quan hóa dữ liệu** không chỉ là công cụ mà còn là cầu nối, mang lại cái nhìn rõ ràng và sâu sắc hơn về thông tin, hỗ trợ tối đa cho các quyết định dựa trên dữ liệu.

### 1.2.1.2 Áp dụng với dataset

**Bước 1:** Đọc và kiểm tra thông tin cơ bản của tập dữ liệu

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     10000 non-null  int64
1   category              10000 non-null  object
2   title                 10000 non-null  object
3   body                 10000 non-null  object
4   amenities             6451 non-null   object
5   bathrooms            9966 non-null   float64
6   bedrooms             9993 non-null   float64
7   currency             10000 non-null  object
8   fee                  10000 non-null  object
9   has_photo            10000 non-null  object
10  pets_allowed         5837 non-null   object
11  price                10000 non-null  int64
12  price_display        10000 non-null  object
13  price_type           10000 non-null  object
14  square_feet          10000 non-null  int64
15  address              6673 non-null   object
16  cityname             9923 non-null   object
17  state                9923 non-null   object
18  latitude             9990 non-null   float64
19  longitude            9990 non-null   float64
20  source               10000 non-null  object
21  time                 10000 non-null  int64
dtypes: float64(4), int64(4), object(14)
memory usage: 1.7+ MB
```

Hình 1-1 Mô tả thông tin dataset

Dữ liệu có 10000 hàng, 22 cột gồm 3 kiểu dữ liệu int64, object, float64 và có các dữ liệu giá trị null.

## Bước 2: Xử lý dữ liệu

Chuyển đổi các giá trị None của thuộc tính **pets\_allowed**.

pets_allowed		count	
		pets_allowed	
Cats,Dogs		5228	
Cats		485	
Dogs		124	
		No	4163
		Cats	485
		Dogs	124

Hình 1-2 Trước và sau xử lý pets\_allowed

Chuyển đổi các giá trị **None** cityname và state theo **kinh độ** và **vĩ độ**.

longitude latitude		count	
		ø	
-98.5576	39.8163	66	cityname 0
-82.1971	28.4590	1	state 0

Hình 1-3 Trước và sau xử lý cityname và state

Chuyển đổi thuộc tính **amenities** thành **Basic** khi nó **None** và **Luxury** khi nó có chứa các giá trị "Gym", "Pool", "Clubhouse", "Hot Tub", "Doorman", "Gated", "View", "Tennis", "Elevator", "Fireplace", "Basketball Court", "Basketball", "Playground".

amenities		count
Parking		229
Dishwasher,Refrigerator		225
Pool		168
Dishwasher,Parking,Pool		149
Dishwasher		147
amenities		count
basic		5635
luxury		4365

Hình 1-4 Trước và sau khi xử lý amenities

Sau đó xóa các hàng có giá trị **None**

	0	pets_allowed	0
id	0	price	0
category	0	price_display	0
title	0	price_type	0
body	0	square_feet	0
amenities	0	cityname	0
bathrooms	0	state	0
bedrooms	0	latitude	0
currency	0	longitude	0
fee	0	source	0
has_photo	0	time	0

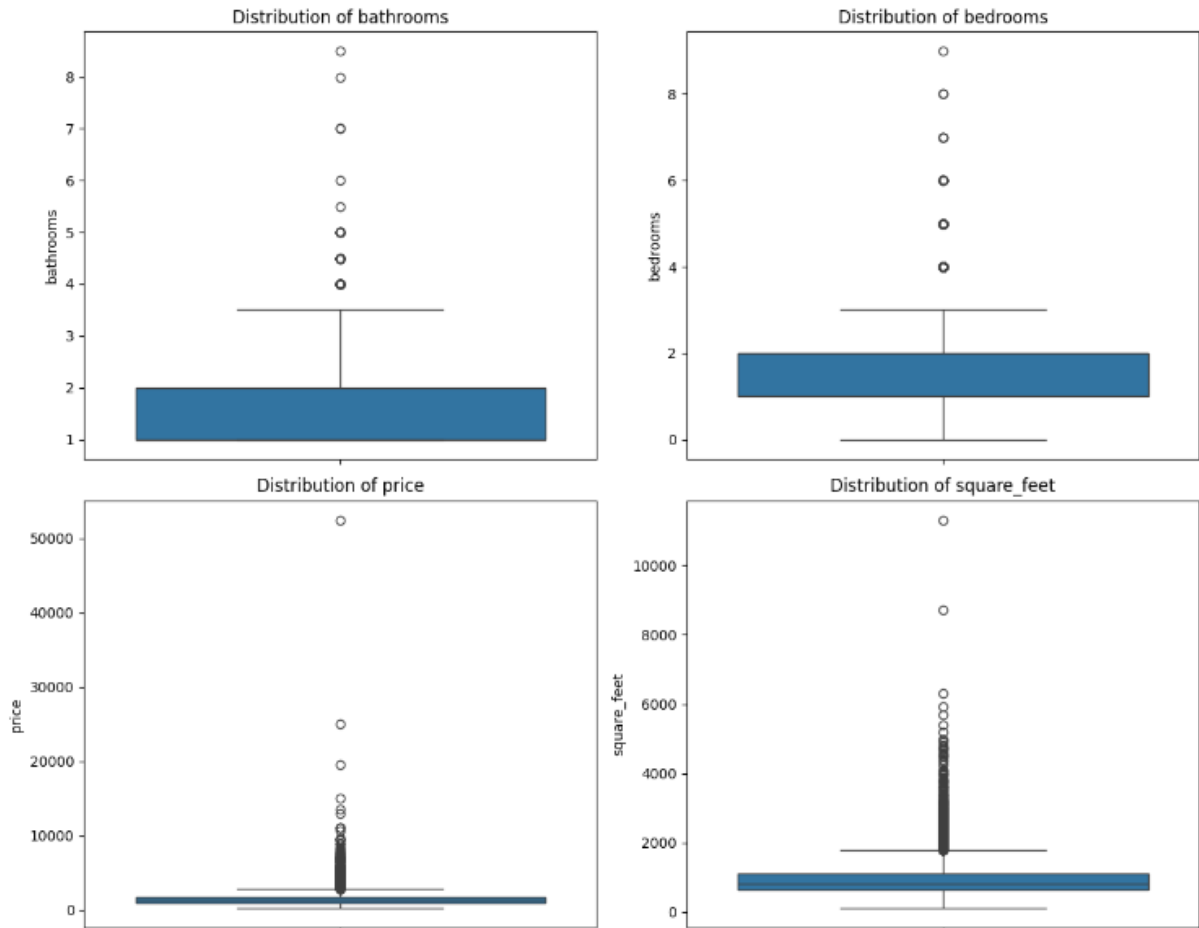
Hình 1-5 Các giá trị NaN của dữ liệu sau khi xử lý



### Bước 3: Trực quan hóa dữ liệu

- Biểu đồ hộp đối với giá trị numerical

Numerical Variables Distribution - Box Plots

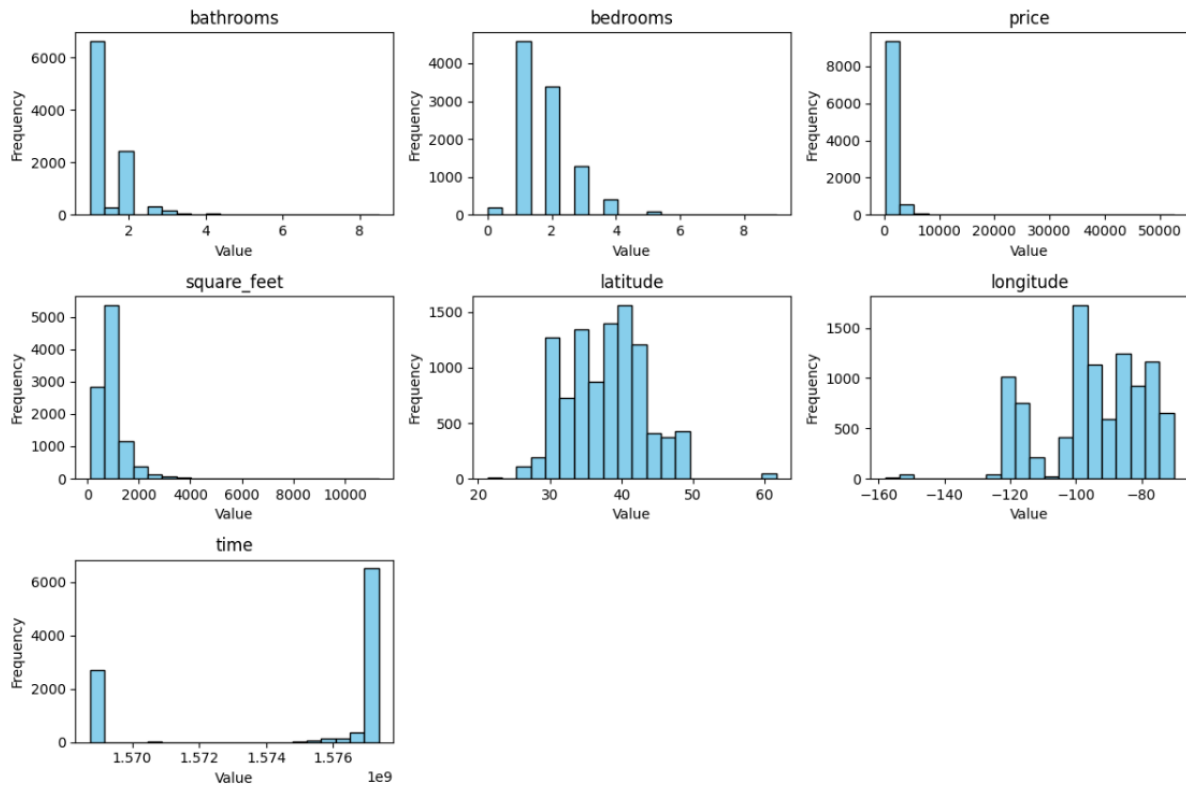


Hình 1-6 Mô tả giá trị numerical của bathrooms, bedrooms, price, square\_feet

Cả bốn phân bố đều có ngoại lai, với số phòng tắm và phòng ngủ có ít giá trị cực đoan hơn so với giá và diện tích.

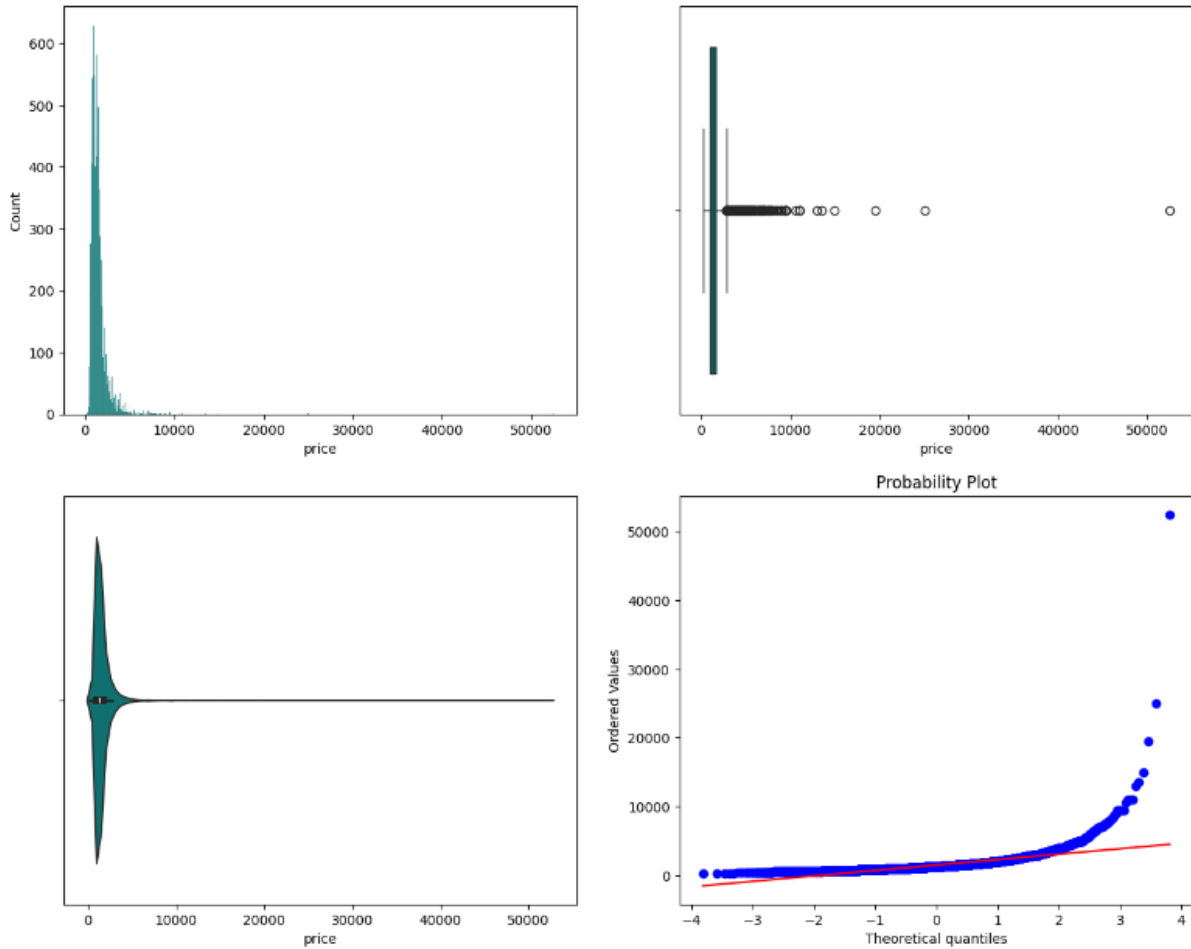
Các giá trị trung vị (đường nằm giữa hộp của mỗi biểu đồ) cho thấy phần lớn bất động sản có số phòng tắm và phòng ngủ ít, giá thấp và diện tích nhỏ. Với một vài bất động sản lớn hoặc đắt hơn làm lệch phân bố.

- Biểu đồ tần suất đối với giá trị numerical



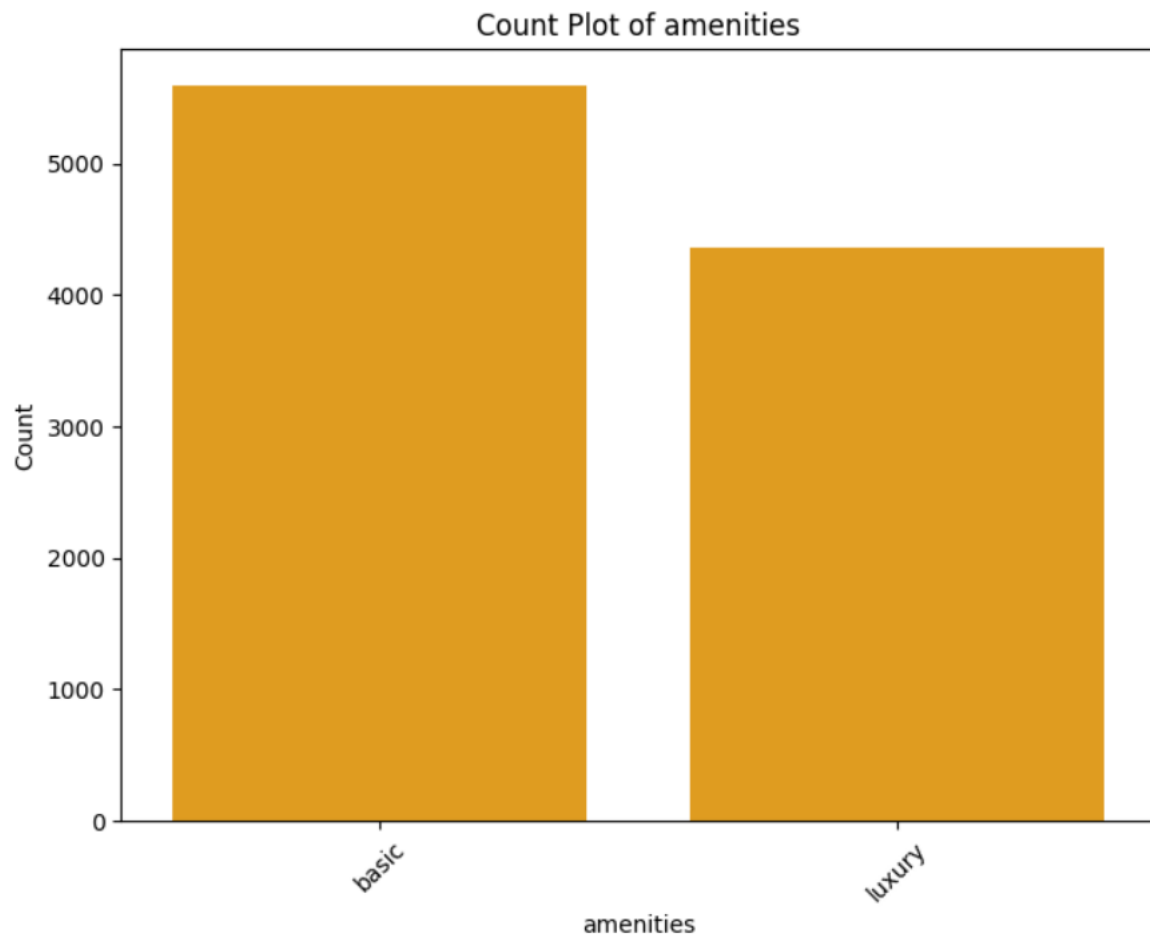
Hình 1-7 Tần suất của các giá trị numerical

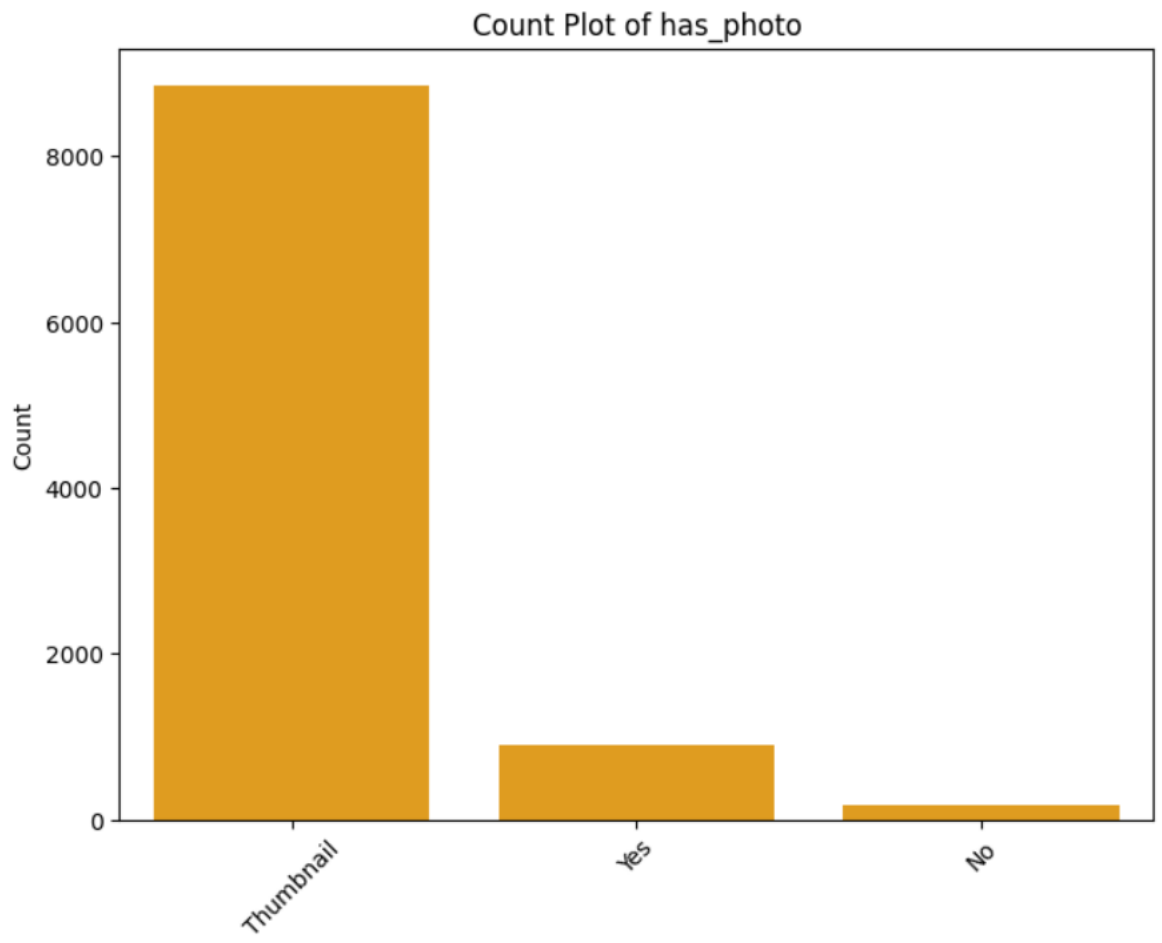
- Đối với price (biểu đồ phân phối, biểu đồ hộp, biểu đồ violin, biểu đồ xác suất chuẩn)

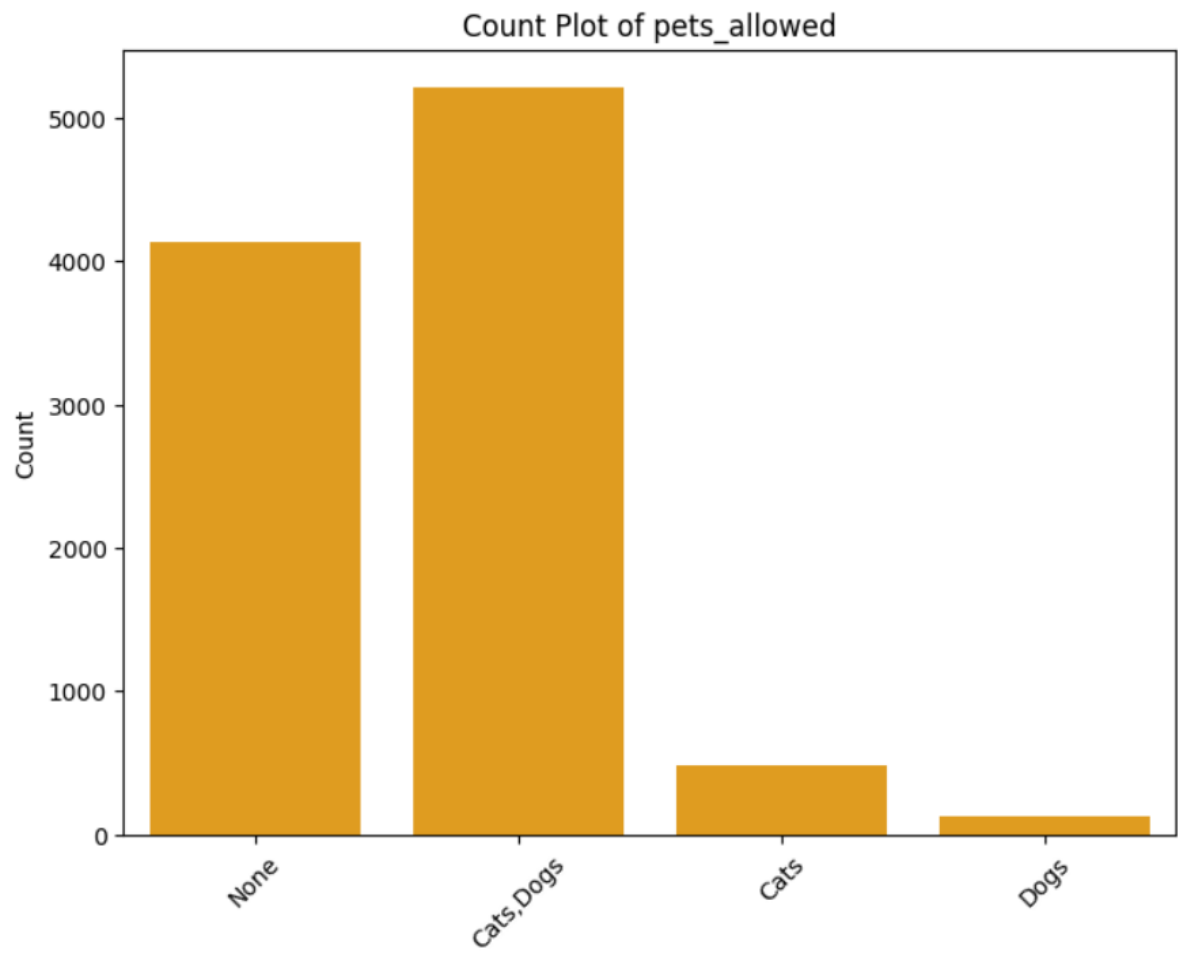


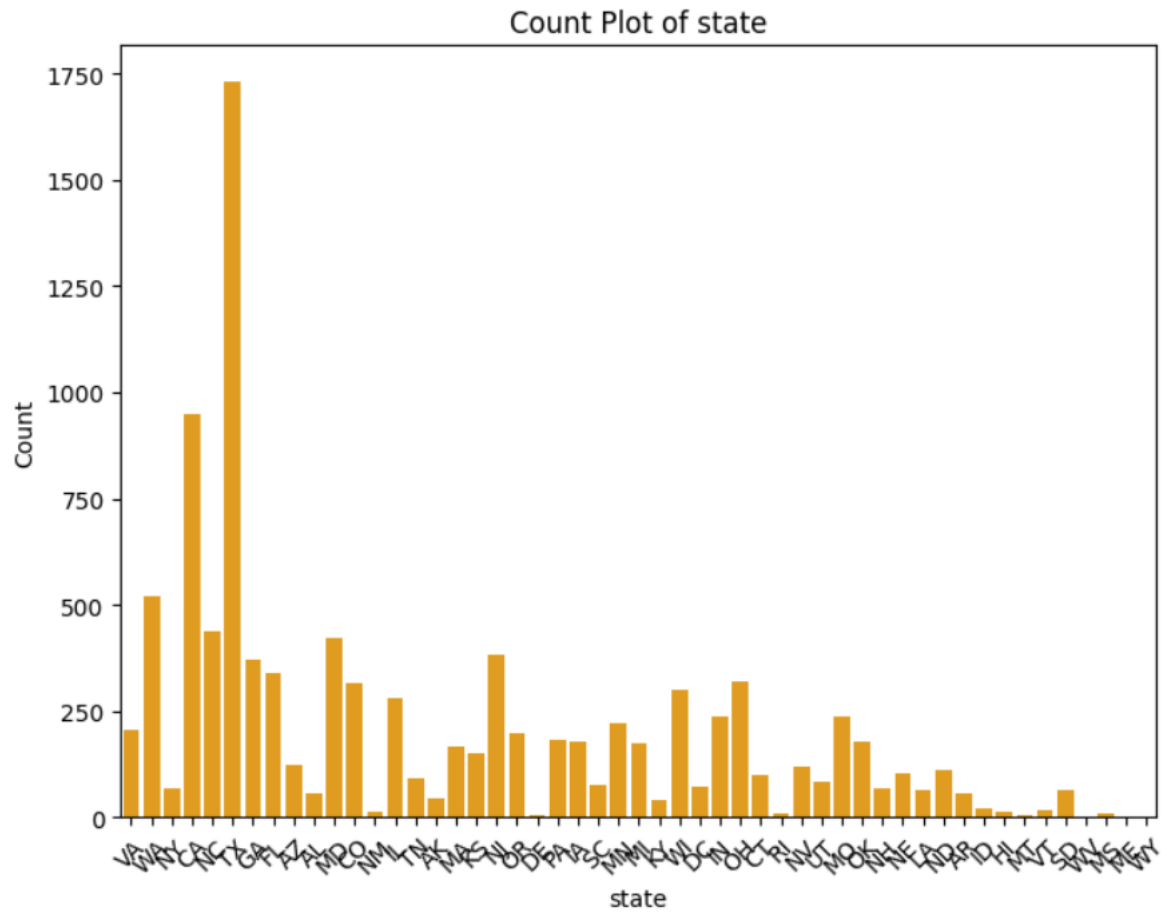
Hình 1-8 Phân phối của price

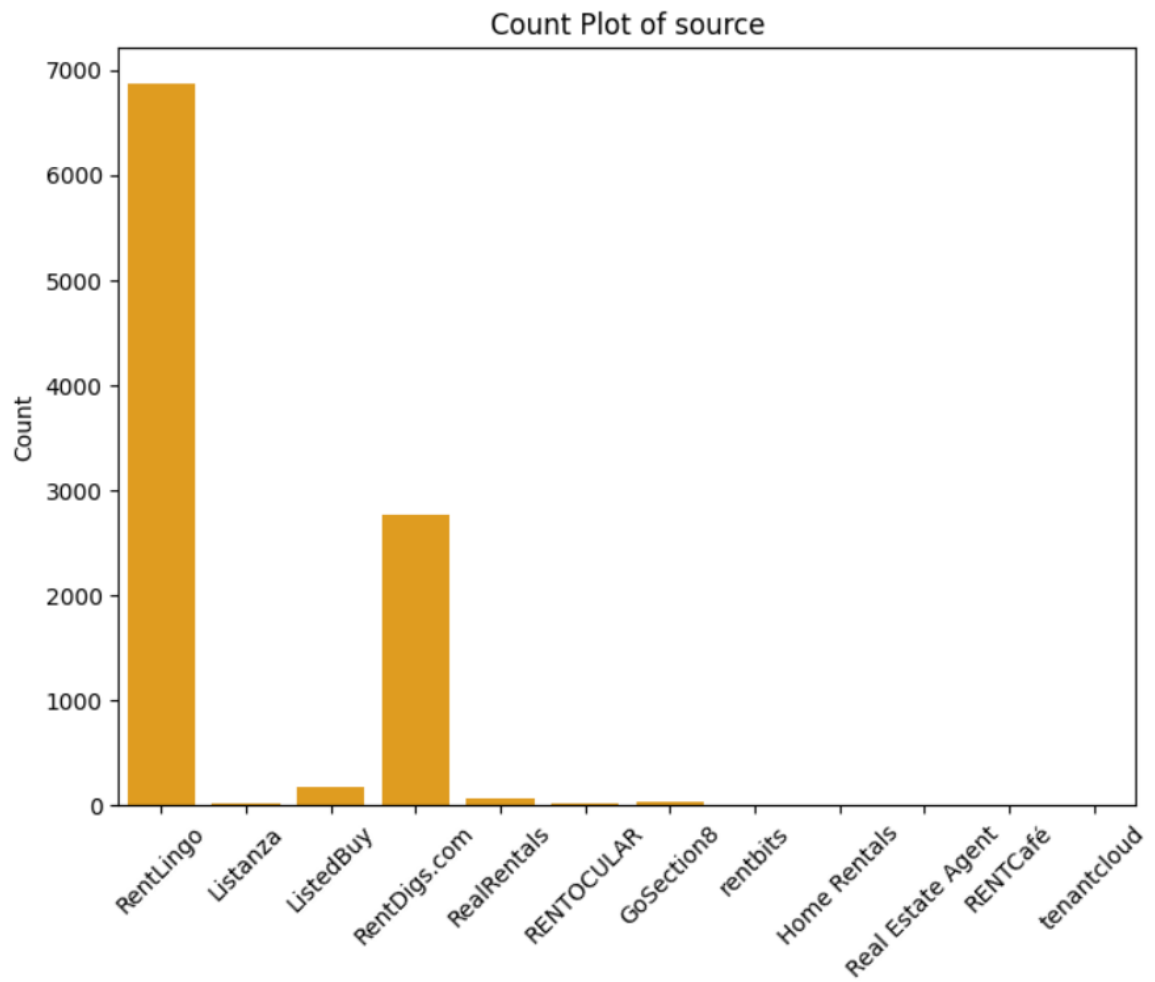
- Biểu đồ cột đối với thuộc tính categorical







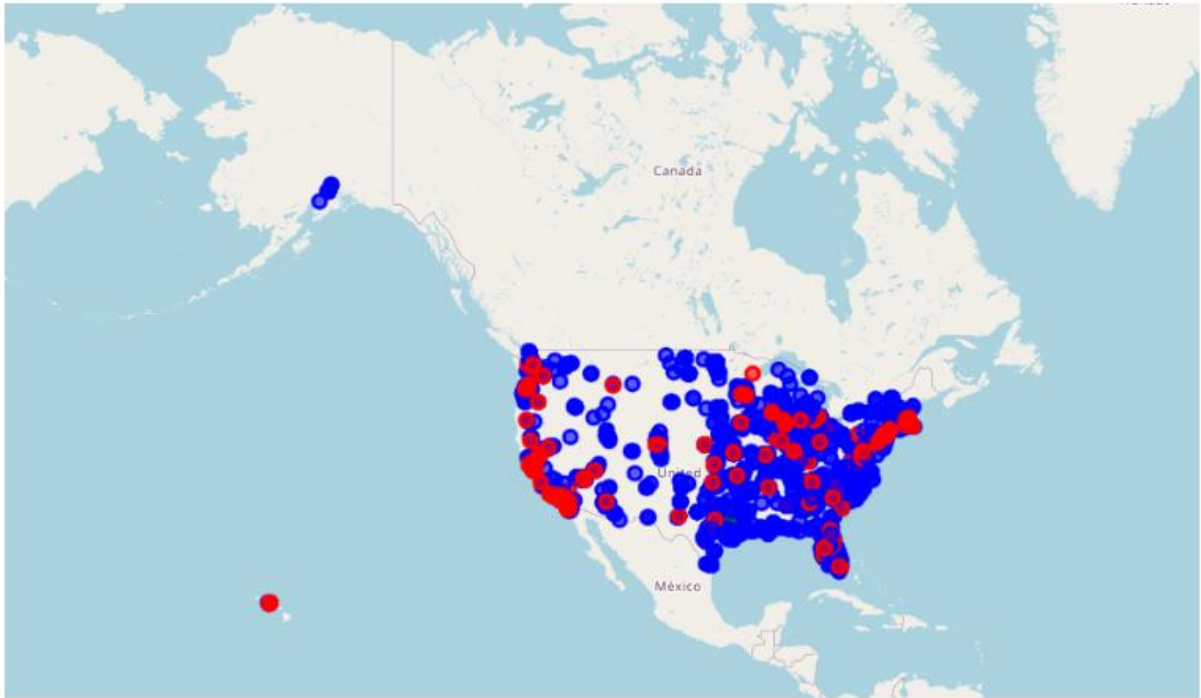




Hình 1-9 Mô tả số lượng phần tử của các thuộc tính

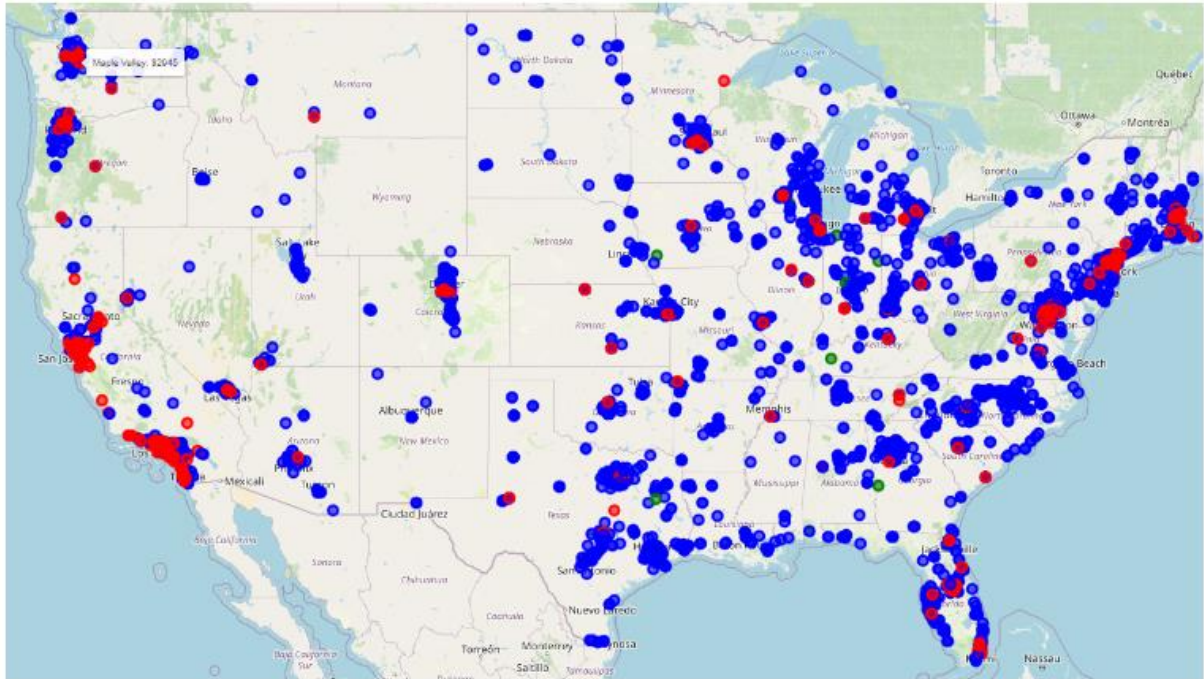


- Phân bố giá nhà theo kinh độ và vĩ độ



Hình 1-10 Mô tả tổng quan phân bố giá nhà

Dữ liệu cho thấy phân bố giá nhà cao tập trung vào các khu vực ven biển hoặc ở các thành phố lớn như **Chicago** và **Losangeles**.



Hình 1-11 Mô tả chi tiết phân bố giá nhà

## 1.2.2 Tiền xử lý dữ liệu

### 1.2.2.1 Type conversion

**Type conversion** là quá trình chuyển đổi các kiểu dữ liệu phân loại (categorical) thành các mã số hoặc các dạng mã hóa số để có thể xử lý trong mô hình học máy. Các biến phân loại thường không có ý nghĩa về mặt số học, nhưng để đưa vào mô hình học máy, ta cần phải chuyển đổi chúng thành dạng số (numerical).

**Phương pháp One-Hot Encoding** hay còn gọi là **Dummies Encoding** là một kỹ thuật xử lý dữ liệu trong học máy và phân tích dữ liệu. Phương pháp này thường được sử dụng để chuyển đổi các biến phân loại (categorical variables) thành dạng số mà mô hình có thể hiểu được. Đối với mỗi giá trị riêng biệt của biến phân loại, One-Hot Encoding sẽ tạo ra một cột mới, và gán giá trị nhị phân 1 nếu hàng đó chứa giá trị tương ứng, và 0 nếu không. Ví dụ, với biến phân loại "Màu sắc" có ba giá trị: "Đỏ", "Xanh" và

"Vàng", phương pháp này sẽ tạo ra ba cột mới tương ứng là "Màu\_Đỏ", "Màu\_Xanh" và "Màu\_Vàng". Sau đó, nếu một dòng dữ liệu có giá trị là "Xanh", cột "Màu\_Xanh" sẽ có giá trị 1, trong khi hai cột còn lại là 0. Phương pháp này giúp loại bỏ sự phụ thuộc thứ tự giữa các giá trị phân loại, vì mỗi giá trị sẽ được đại diện độc lập, mà không có quan hệ thứ bậc. One-Hot Encoding giúp mô hình học máy dễ dàng xử lý và sử dụng dữ liệu phân loại mà không làm mất đi thông tin quan trọng.

#	Column	Non-Null Count	Dtype
0	bathrooms	9949 non-null	float64
1	bedrooms	9949 non-null	float64
2	has_photo	9949 non-null	int64
3	pets_allowed	9949 non-null	int64
4	price	9949 non-null	int64
5	square_feet	9949 non-null	int64
6	latitude	9949 non-null	float64
7	longitude	9949 non-null	float64
8	year	9949 non-null	int32
9	month	9949 non-null	int32
10	day_of_week	9949 non-null	int32
11	hour	9949 non-null	int32
12	is_weekend	9949 non-null	int32
13	amenities_basic	9949 non-null	bool
14	amenities_luxury	9949 non-null	bool
15	source_GoSection8	9949 non-null	bool
16	source_Home Rentals	9949 non-null	bool
17	source_Listanza	9949 non-null	bool
18	source_ListedBuy	9949 non-null	bool
19	source_RENTCafé	9949 non-null	bool
...			
25	source_rentbits	9949 non-null	bool
26	source_tenantcloud	9949 non-null	bool

Hình 1-12 Mô tả kết quả One-Hot Encoding

#### 1.2.2.2 Data normalization

**Chuẩn hóa dữ liệu (Data Normalization)** là quá trình thay đổi quy mô các đặc trưng số trong dữ liệu sao cho các giá trị của chúng nằm trong một phạm vi giống nhau, giúp giảm sự chênh lệch giữa các đặc trưng có đơn vị hoặc phạm vi giá trị khác nhau. Khi các đặc trưng có giá trị lớn hoặc nhỏ

đáng kể, chúng có thể ảnh hưởng không cân đối đến mô hình học máy, đặc biệt là trong những thuật toán phụ thuộc vào khoảng cách như KNN (K-Nearest Neighbors), SVM (Support Vector Machines), hay các mạng nơ-ron (neural networks). Do đó, chuẩn hóa dữ liệu giúp cải thiện hiệu quả và độ chính xác của mô hình.

Một phương pháp chuẩn hóa phổ biến là **MinMaxScaler**, được sử dụng để đưa các giá trị của các đặc trưng về một phạm vi xác định. Thông thường, MinMaxScaler sẽ đưa các giá trị của đặc trưng vào phạm vi từ **0 đến 1**, nhưng người dùng cũng có thể tùy chỉnh phạm vi này theo yêu cầu (ví dụ: từ -1 đến 1). Phương pháp này hoạt động bằng cách sử dụng công thức:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Trong đó:

- $X$  là giá trị gốc của đặc trưng.
- $X_{\min}$  và  $X_{\max}$  lần lượt là giá trị nhỏ nhất và lớn nhất của đặc trưng trong tập dữ liệu.
- $X_{\text{scaled}}$  là giá trị sau khi chuẩn hóa.

Nhờ vào phương pháp này, tất cả các đặc trưng có thể có các giá trị nằm trong cùng một phạm vi, giúp giảm thiểu sự ảnh hưởng không cân đối của các đặc trưng có giá trị lớn (ví dụ: lương, chiều cao) so với các đặc trưng có giá trị nhỏ (ví dụ: tuổi, số lần mua). Đồng thời, **MinMaxScaler** vẫn giữ lại thông tin về phân bố dữ liệu ban đầu, do đó, các đặc trưng không bị mất thông tin quan trọng khi được chuẩn hóa.

**MinMaxScaler** có thể không phù hợp trong các trường hợp dữ liệu có nhiều giá trị ngoại lệ (outliers), vì các ngoại lệ này sẽ ảnh hưởng lớn đến phạm vi giá trị của toàn bộ đặc trưng. Trong những trường hợp này, các phương pháp chuẩn hóa khác như StandardScaler có thể được sử dụng.

## 1.3 Giới thiệu mô hình và tham số

### 1.3.1 Giải thích mô hình

#### 1.3.1.1 Linear Regression

- **Khái niệm:**

Hồi quy tuyến tính là một trong những mô hình đơn giản và phổ biến nhất trong học máy, thường được áp dụng khi có giả định về quan hệ tuyến tính giữa biến độc lập  $X$  và biến phụ thuộc  $y$ . Mô hình này cố gắng tìm phương trình của một đường thẳng (hoặc mặt phẳng, trong trường hợp nhiều biến) để dự đoán giá trị của  $y$  dựa trên  $X$ . Công thức của hồi quy tuyến tính đơn biến là:

$$y = w.x + b$$

Trong đó:  $w$  là hệ số hồi quy (độ dốc) và  $b$  là hằng số.

- **Ứng dụng thực tiễn của Linear Regression:**

Linear Regression (hồi quy tuyến tính) là một thuật toán học máy đơn giản nhưng rất hiệu quả, được áp dụng rộng rãi trong các lĩnh vực như kinh tế, tài chính, y học, và kỹ thuật. Một số ứng dụng phổ biến bao gồm dự báo doanh thu, dự đoán giá bất động sản, phân tích xu hướng thị trường, và đánh giá tác động của các yếu tố khác nhau lên kết quả kinh doanh. Chẳng hạn, trong kinh doanh, Linear Regression có thể được sử dụng để dự đoán doanh số bán hàng dựa trên ngân sách quảng cáo, hoặc dự đoán giá cổ phiếu dựa trên các yếu tố kinh tế vĩ mô. Trong y học, Linear Regression giúp phân tích mối quan hệ giữa các chỉ số sức khỏe (như huyết áp, chỉ số BMI) và khả năng mắc các bệnh tim mạch.

- **Ưu điểm của Linear Regression**

**Đơn giản và dễ hiểu:** Linear Regression là một trong những thuật toán đơn giản nhất trong học máy, dễ dàng triển khai và giải thích. Kết quả

của nó là một đường thẳng, giúp phân tích và hiểu mối quan hệ giữa các biến trở nên rõ ràng.

**Tính toán nhanh:** Với công thức đơn giản, Linear Regression yêu cầu ít tài nguyên tính toán hơn nhiều so với các thuật toán phức tạp khác, do đó phù hợp cho việc triển khai trên các hệ thống có tài nguyên hạn chế.

**Khả năng diễn giải hệ số:** Linear Regression cung cấp các hệ số hồi quy, cho phép đánh giá ảnh hưởng của từng biến độc lập đến biến phụ thuộc, giúp ích cho việc phân tích và ra quyết định.

- **Nhược điểm của Linear Regression**

**Giả định về tuyến tính:** Linear Regression chỉ mô hình hóa các mối quan hệ tuyến tính, do đó không hiệu quả khi mối quan hệ giữa biến độc lập và biến phụ thuộc không tuyến tính. Trong các trường hợp phức tạp, các mô hình phi tuyến tính sẽ cho kết quả tốt hơn.

**Nhạy cảm với outliers:** Linear Regression rất nhạy cảm với các giá trị ngoại lệ. Chỉ một vài điểm dữ liệu khác biệt quá xa có thể làm thay đổi đường hồi quy và ảnh hưởng lớn đến dự đoán.

**Giả định về phân phối chuẩn và phương sai không đổi:** Linear Regression giả định rằng các biến độc lập phải tuân theo phân phối chuẩn và có phương sai không đổi (homoscedasticity). Khi các giả định này không thỏa mãn, mô hình có thể cho kết quả không chính xác hoặc có hiệu quả kém.

#### 1.3.1.2 Random Forest Regressor

- **Khái niệm**

Random Forest là một thuật toán học máy thuộc nhóm các phương pháp Ensemble Learning, có thể được sử dụng cho cả bài toán hồi quy và phân loại. Random Forest Regressor là một phiên bản của thuật toán này dành riêng cho các bài toán hồi quy, nhằm dự đoán giá trị số liên tục. Thuật toán hoạt động dựa trên việc kết hợp nhiều cây quyết định (Decision Trees)

để đưa ra một dự đoán mạnh mẽ và đáng tin cậy hơn so với một cây quyết định đơn lẻ.

- **Cấu trúc và cơ chế hoạt động của Random Forest**

**Random Forest** là một tập hợp của nhiều cây quyết định, mỗi cây hoạt động độc lập và đưa ra dự đoán riêng biệt. Mỗi cây trong rừng được huấn luyện trên một mẫu ngẫu nhiên từ dữ liệu ban đầu, gọi là phương pháp Bootstrap Aggregation (hoặc Bagging). Mô hình Random Forest Regressor sử dụng các bước sau:

- **Bootstrap Sampling:** Lấy nhiều mẫu ngẫu nhiên từ tập dữ liệu huấn luyện, mỗi mẫu có kích thước bằng với tập huấn luyện và được lấy ngẫu nhiên với thay thế (tức là mỗi mẫu dữ liệu có thể xuất hiện nhiều lần trong cùng một tập mẫu).
  - **Xây dựng các cây quyết định:** Đối với mỗi mẫu bootstrap, một cây quyết định được xây dựng. Tại mỗi nút của cây, thuật toán chỉ chọn ngẫu nhiên một số lượng đặc trưng nhất định (thường là căn bậc hai của tổng số đặc trưng) để tìm ra đặc trưng nào tốt nhất để phân chia, thay vì xét tất cả các đặc trưng như trong cây quyết định thông thường. Điều này làm cho các cây khác biệt và giảm sự phụ thuộc vào một số đặc trưng nhất định.
  - **Tập hợp dự đoán:** Trong bài toán hồi quy, đầu ra của Random Forest là giá trị trung bình dự đoán của tất cả các cây trong rừng. Mỗi cây dự đoán một giá trị cho một mẫu dữ liệu đầu vào, và giá trị cuối cùng là trung bình của tất cả các dự đoán này, giúp giảm thiểu sai số và tạo ra dự đoán ổn định hơn.
- **Ưu điểm của Random Forest Regressor**

**Giảm thiểu hiện tượng overfitting:** Random Forest Regressor kết hợp nhiều cây quyết định, giúp mô hình có khả năng khái quát hóa tốt hơn so với một cây quyết định đơn lẻ, đặc biệt khi dữ liệu có nhiễu.

**Khả năng đánh giá quan trọng của đặc trưng:** Random Forest cung cấp thông tin về tầm quan trọng của các đặc trưng trong quá trình dự đoán, hữu ích khi cần thực hiện chọn lọc đặc trưng.

**Khả năng mở rộng:** Mô hình có thể mở rộng để phù hợp với các dữ liệu có kích thước lớn mà không yêu cầu điều chỉnh nhiều tham số.

- **Nhược điểm của Random Forest Regressor**

**Chi phí tính toán cao:** Random Forest sử dụng nhiều cây quyết định, nên quá trình huấn luyện và dự đoán có thể tốn nhiều thời gian và tài nguyên tính toán, đặc biệt khi dữ liệu lớn.

**Giảm khả năng diễn giải:** Với một cây quyết định đơn lẻ, việc diễn giải kết quả dễ dàng hơn, tuy nhiên với một rừng cây, việc hiểu rõ cách mỗi đặc trưng ảnh hưởng đến kết quả dự đoán trở nên phức tạp hơn.

**Không hiệu quả với dữ liệu có nhiều giá trị ngoại lệ:** Mặc dù Random Forest ít nhạy cảm với outliers hơn so với các mô hình tuyến tính, các giá trị ngoại lệ vẫn có thể gây ảnh hưởng nếu có quá nhiều.

### 1.3.1.3 Decision Tree Regressor

- **Khái niệm**

**Decision Tree Regressor** là một cấu trúc cây, trong đó mỗi nút trong cây đại diện cho một đặc trưng của dữ liệu, mỗi nhánh đại diện cho một giá trị hoặc phạm vi giá trị của đặc trưng đó, và mỗi lá của cây đại diện cho một giá trị dự đoán. Cây quyết định được xây dựng bằng cách đệ quy chia nhỏ dữ liệu để giảm thiểu độ lệch so với giá trị thực tế tại các lá.

**Quá trình xây dựng Decision Tree Regressor:**



**Chia nhỏ dữ liệu:** Decision Tree chia dữ liệu thành các nhánh dựa trên đặc trưng và giá trị để tối đa hóa độ thuần khiết trong mỗi nhánh.

**Điều kiện dừng:** Quá trình phân chia tiếp tục cho đến khi đạt một điều kiện dừng nào đó, ví dụ như đạt đến độ sâu tối đa của cây, số lượng mẫu tối thiểu trong một nút, hoặc lỗi dự đoán nhỏ.

Cây quyết định đưa ra dự đoán cuối cùng bằng cách đi theo các nhánh dựa trên các điều kiện của từng nút cho đến khi đến một lá. Giá trị tại lá đó là giá trị dự đoán của mô hình.

- **Ưu điểm của Decision Tree Regressor**

**Dễ hiểu và giải thích:** Decision Tree Regressor tạo ra một cấu trúc cây dễ hiểu và trực quan, giúp việc giải thích các quyết định trở nên đơn giản. Người dùng có thể dễ dàng thấy cách thức mà mô hình đưa ra dự đoán.

**Xử lý tốt dữ liệu phức tạp:** Decision Tree Regressor có khả năng xử lý dữ liệu với các đặc trưng không đồng nhất hoặc phi tuyến tính mà không cần biến đổi hoặc chuẩn hóa dữ liệu đầu vào.

**Linh hoạt và mạnh mẽ:** Decision Tree có thể xử lý cả dữ liệu có nhiễu và dữ liệu không hoàn hảo, và có thể được áp dụng cho các tập dữ liệu có tính chất đa dạng.

- **Nhược điểm của Decision Tree Regressor**

**Dễ bị overfitting:** Nếu không giới hạn chiều sâu của cây, Decision Tree Regressor dễ dàng khớp quá sát với dữ liệu huấn luyện, dẫn đến overfitting và giảm hiệu quả khi áp dụng trên dữ liệu mới.

**Nhạy cảm với dữ liệu nhiễu:** Một vài giá trị bất thường có thể gây ảnh hưởng lớn đến cấu trúc của cây, dẫn đến dự đoán sai lệch.

**Kém hiệu quả trên các bài toán phức tạp:** Khi mô hình cần dự đoán trên các dữ liệu phức tạp hoặc có mối quan hệ phi tuyến tính cao, Decision

Tree Regressor thường có độ chính xác kém hơn so với các phương pháp ensemble như Random Forest hoặc Gradient Boosting.

#### 1.3.1.4 Gradient Boosting Regressor

- **Khái niệm**

**Gradient Boosting Regressor** là một thuật toán học máy thuộc nhóm các phương pháp Ensemble Learning, được sử dụng chủ yếu cho các bài toán hồi quy nhằm dự đoán các giá trị liên tục. Gradient Boosting hoạt động bằng cách kết hợp tuần tự nhiều mô hình (các cây quyết định nhỏ gọi là "weak learners") lại với nhau, mỗi mô hình mới sẽ tập trung vào việc sửa các lỗi của mô hình trước đó. Ý tưởng cốt lõi là sử dụng phương pháp "boosting" để tối ưu dần dần hiệu suất của mô hình, cải thiện độ chính xác tổng thể trong việc dự đoán.

- **Ưu điểm của Gradient Boosting Regressor**

**Hiệu quả cao trong việc dự đoán chính xác:** Gradient Boosting Regressor thường cho kết quả rất chính xác trong các bài toán hồi quy phức tạp, vì nó có khả năng tối ưu lỗi rất hiệu quả.

**Xử lý được dữ liệu có đặc trưng phi tuyến tính:** Mô hình có thể xử lý tốt dữ liệu với các mối quan hệ phi tuyến tính và phức tạp giữa các đặc trưng.

**Tính linh hoạt cao:** Gradient Boosting có thể sử dụng với các hàm mất mát khác nhau, cho phép ứng dụng vào nhiều loại bài toán khác nhau và tối ưu hóa phù hợp với từng yêu cầu cụ thể.

- **Nhược điểm của Gradient Boosting Regressor**

**Dễ bị overfitting:** Gradient Boosting có khả năng overfit, đặc biệt khi số lượng cây quá lớn hoặc không điều chỉnh cẩn thận các tham số như learning rate hoặc độ sâu của cây.

**Thời gian huấn luyện lâu:** Do mỗi cây phụ thuộc vào cây trước đó và phải tính toán gradient nhiều lần, Gradient Boosting đòi hỏi tài nguyên tính toán cao và mất nhiều thời gian hơn so với các mô hình như Decision Tree hay Random Forest.

**Khó giải thích hơn các mô hình đơn giản:** Do cấu trúc phức tạp của nhiều cây kết hợp với nhau, Gradient Boosting khó diễn giải hơn các mô hình như Linear Regression hoặc Decision Tree đơn lẻ.




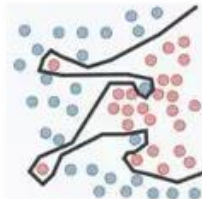
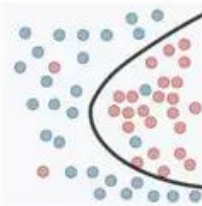
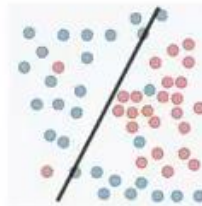

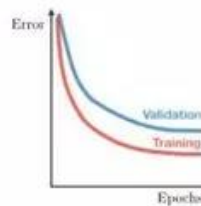

### 1.3.2 Giải thích metric

Overfitting và Underfitting là hai nguyên nhân lớn nhất dẫn đến hiệu suất kém của các thuật toán học máy.

**Overfitting:** Xảy ra khi Mô hình hoạt động tốt đối với một tập hợp dữ liệu cụ thể (Dữ liệu đã biết) và do đó có thể không phù hợp với dữ liệu bổ sung (Dữ liệu không xác định).

**Underfitting:** Xảy ra khi mô hình không thể nắm bắt đầy đủ cấu trúc cơ bản của dữ liệu.

**Tổng quát hóa:** đề cập đến mức độ áp dụng của các khái niệm được học bởi một mô hình học máy đối với các ví dụ cụ thể mà mô hình không nhìn thấy khi nó đang học.

	Overfitting	Generalization	Underfitting
Regression Illustration			
Classification Illustration			
Deep Learning Illustration			

Hình 1-13 Mô tả Overfitting và Underfitting

### 1.3.2.1 Mean Absolute Error (MAE) - Sai số trung bình tuyệt đối

- **Công thức**

**Mean Absolute Error (MAE)** đo độ lớn trung bình của các lỗi trong một tập hợp các dự đoán mà không cần xem xét hướng của chúng. Đó là giá trị trung bình trên mẫu thử nghiệm về sự khác biệt tuyệt đối giữa dự đoán và quan sát thực tế, trong đó tất cả các khác biệt riêng lẻ có trọng số bằng nhau.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

trong đó  $n$  là số điểm dữ liệu,  $y_i$  là giá trị quan sát và  $\hat{y}_i$  là giá trị dự đoán.

Có thể diễn đạt MAE là tổng hòa của hai thành phần: Bất đồng về số lượng và Bất đồng về phân bố.

MAE được biết đến là mạnh mẽ hơn đối với các yếu tố ngoại lai so với MSE. Lý do chính là trong MSE bằng cách bình phương các sai số, các giá trị ngoại lai (thường có sai số cao hơn các mẫu khác) được chú ý nhiều hơn và chiếm ưu thế trong sai số cuối cùng và tác động đến các tham số của mô hình.

MAE càng thấp thì dự báo càng tốt.

#### **Áp dụng**

MAE đặc biệt hữu ích trong ngữ cảnh dự đoán diện tích vì nó thể hiện sai số theo đơn vị gốc (square feet), giúp dễ hiểu về độ chính xác của mô hình khi dự đoán diện tích. Ví dụ, nếu MAE là 50, điều đó có nghĩa là dự đoán của mô hình lệch trung bình 50 square feet so với diện tích thực tế.

### 1.3.2.2 Mean Absolute Error (MAE) - Sai số trung bình tuyệt đối

**Root Mean Square Error (RMSE) hoặc Root Mean Square Deviation (RMSD)** là căn bậc hai của mức trung bình của các sai số bình phương. RMSE là độ lệch chuẩn của các phần dư (sai số dự đoán).

Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ dàn trải của những phần dư này, nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh đường phù hợp nhất.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}}$$

**RMSE** = root-mean-square deviation

$i$  = variable  $i$

$N$  = number of non-missing data points

$x_i$  = actual observations time series

$\hat{x}_i$  = estimated time series

Ảnh hưởng của mỗi lỗi đối với RMSE tỷ lệ với kích thước của lỗi bình phương; do đó các sai số lớn hơn có ảnh hưởng lớn đến RMSE một cách không cân xứng. Do đó, RMSE nhạy cảm với các yếu tố ngoại lai. Sai số bình phương trung bình gốc thường được sử dụng trong khí hậu học, dự báo và phân tích hồi quy để xác minh kết quả thực nghiệm.

Khi các quan sát và dự báo chuẩn hóa được sử dụng làm đầu vào RMSE, có mối quan hệ trực tiếp với hệ số tương quan. Ví dụ, nếu hệ số tương quan là 1, RMSE sẽ bằng 0, bởi vì tất cả các điểm nằm trên đường hồi quy (và do đó không có sai số).

RMSE luôn không âm và giá trị 0 (hầu như không bao giờ đạt được trong thực tế) sẽ chỉ ra sự phù hợp hoàn hảo với dữ liệu. Nói chung, RMSD thấp hơn sẽ tốt hơn RMSD cao hơn.

### 1.3.2.3 Accuracy

#### **Số liệu phân loại**

Khi thực hiện các dự đoán phân loại, có bốn loại kết quả có thể xảy ra:

- True positives (TP) là khi bạn dự đoán một quan sát thuộc về một lớp và nó thực sự thuộc về lớp đó.
- True Negative (TN) là khi bạn dự đoán một quan sát không thuộc về một lớp và nó thực sự không thuộc lớp đó.
- False Positive (FP) xảy ra khi bạn dự đoán một quan sát thuộc về một lớp trong khi thực tế thì không.
- False Negative (FN) xảy ra khi bạn dự đoán một quan sát không thuộc về một lớp trong khi thực tế là nó có.

Bốn kết quả này thường được vẽ trên một ma trận nhầm lẫn. Ma trận nhầm lẫn sau đây là một ví dụ cho trường hợp phân loại nhị phân. Bạn sẽ tạo ma trận này sau khi đưa ra dự đoán trên dữ liệu thử nghiệm của mình và sau đó xác định từng dự đoán là một trong bốn kết quả có thể được mô tả ở trên.

		Prediction	
		0	1
True Label	0	48 true negatives	8 false positives
	1	4 false negatives	37 true positives

Hình 1-14 Ma trận nhầm lẫn

Accuracy là một thước đo để đánh giá các mô hình phân loại. Về mặt hình thức, accuracy có thể được định nghĩa là số lần dự đoán đúng trên tổng số lần dự đoán.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Viết điều này theo True Positive và True Negative: Tỷ lệ phần trăm trường hợp tích cực trong tổng số trường hợp tích cực thực tế. Do đó, mẫu số ( TP + FN) ở đây là số lượng thực tế các trường hợp dương có trong tập dữ liệu.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$



## 1.4 Đánh giá và so sánh kết quả

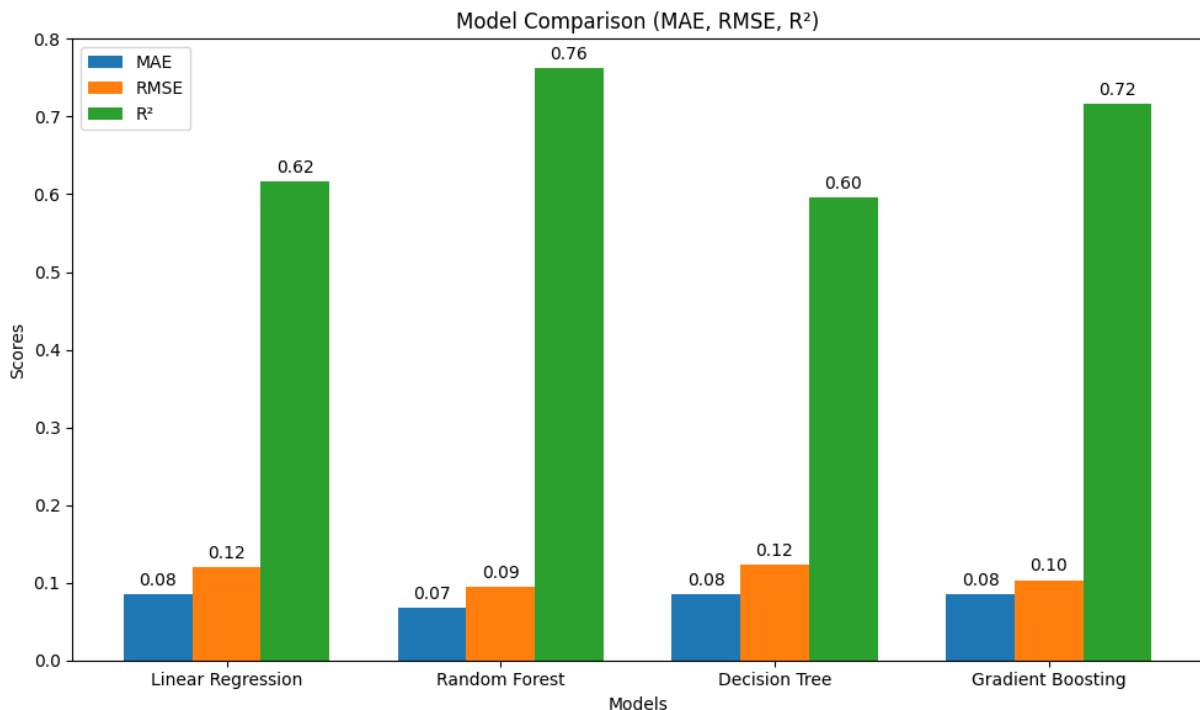
### 1.4.1 Phân chia dữ liệu

Việc phân chia dữ liệu thành các bộ huấn luyện và kiểm tra giúp đánh giá hiệu quả của mô hình trên dữ liệu mà nó chưa được học. Sau khi thực hiện phân chia, các kết quả trả về bao gồm:

- $X_{train}$ : Dữ liệu đặc trưng của bộ huấn luyện
- $X_{test}$ : Dữ liệu đặc trưng của bộ kiểm tra
- $y_{train}$ : Nhãn mục tiêu của bộ huấn luyện
- $y_{test}$ : Nhãn mục tiêu của bộ kiểm tra

### 1.4.2 Hồi quy

#### 1.4.2.1 So sánh các chỉ số MAE, RMSE, $R^2$



Hình 1-15 So sánh MAE, RMSE,  $R^2$

### **Đánh giá:**

**Linear Regression:** Linear Regression có MAE và RMSE tương đối nhỏ, cho thấy mô hình có khả năng dự đoán khá chính xác. Tuy nhiên,  $R^2$  chỉ đạt 0.62, tức là mô hình chỉ giải thích được 62% biến thiên của dữ liệu, điều này cho thấy nó có thể bỏ lỡ một phần thông tin quan trọng.

**Random Forest:** Random Forest có MAE và RMSE thấp nhất trong các mô hình, cho thấy nó có độ chính xác tốt nhất trong việc dự đoán diện tích. Hơn nữa,  $R^2$  đạt 0.76, cao nhất trong các mô hình, nghĩa là Random Forest giải thích được 76% biến thiên của dữ liệu và phù hợp tốt hơn so với các mô hình khác.

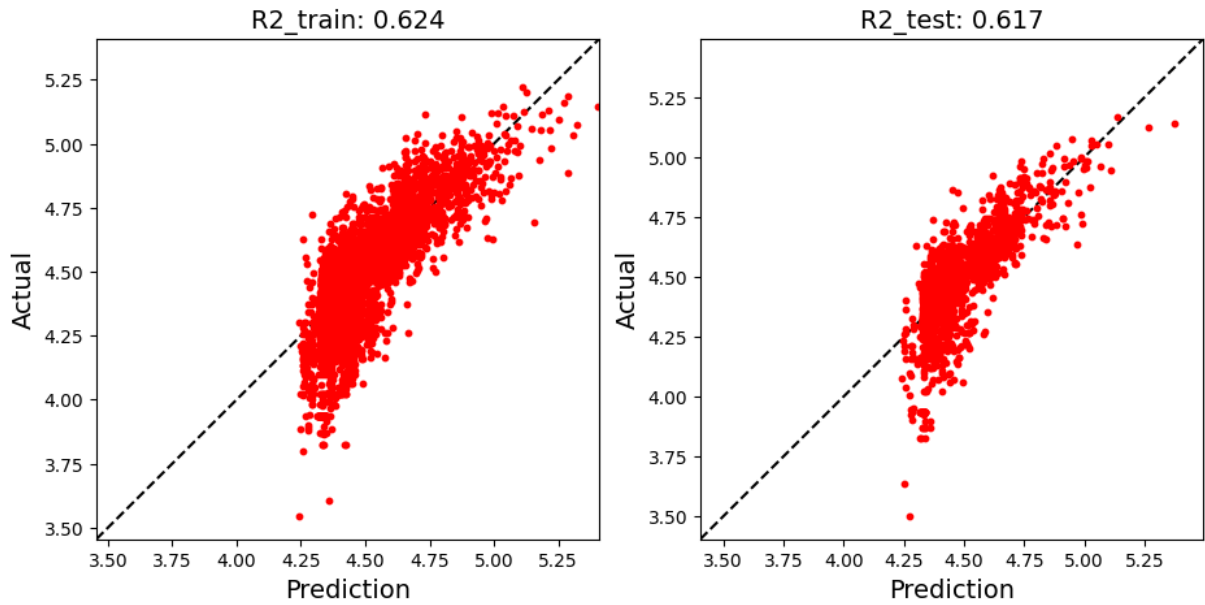
**Decision Tree:** Decision Tree có MAE và RMSE tương đương với Linear Regression nhưng chỉ đạt  $R^2$  0.59, thấp hơn so với các mô hình khác. Điều này cho thấy Decision Tree ít hiệu quả hơn trong việc giải thích biến thiên của dữ liệu, có thể bị ảnh hưởng bởi sự quá khớp (overfitting) hoặc hạn chế trong việc mô hình hóa mối quan hệ phức tạp.

**Gradient Boosting:** Gradient Boosting có MAE và RMSE thấp và  $R^2$  đạt 0.72, cho thấy nó có hiệu suất khá tốt, mặc dù không bằng Random Forest. Gradient Boosting dường như cân bằng giữa độ chính xác của dự đoán và khả năng giải thích biến thiên của dữ liệu.

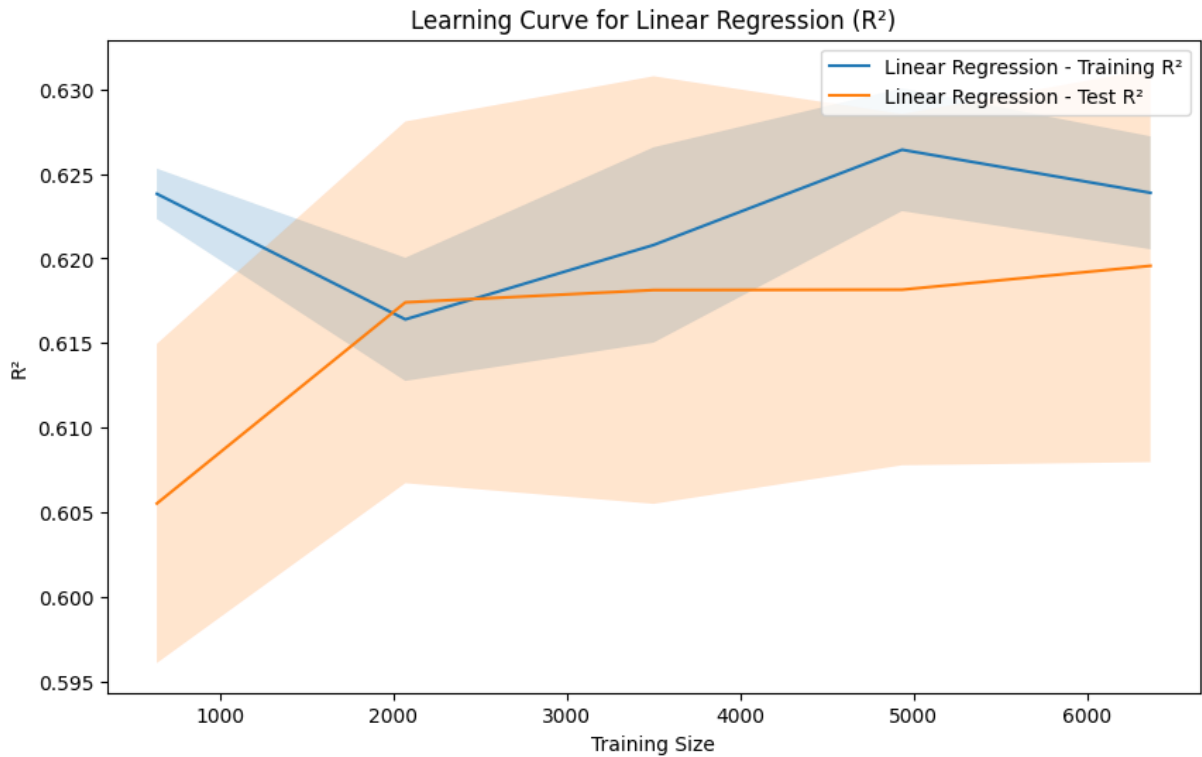
**Nhận xét:** Random Forest có hiệu suất tốt nhất với MAE và RMSE thấp nhất và cao nhất, cho thấy nó phù hợp nhất cho bài toán dự đoán diện tích.

### 1.4.2.2 So sánh Learning Curve

- **Linear Regression:**



Hình 1-16 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Linear Regression



Hình 1-17 Learning Curve của Linear Regression

#### **Đường huấn luyện (Training):**

Trong quá trình huấn luyện, ban đầu, khi kích thước của tập dữ liệu huấn luyện nhỏ, hệ số  $R^2$  đạt giá trị cao (khoảng 0.625). Điều này xảy ra vì mô hình có ít dữ liệu hơn để điều chỉnh và có thể dễ dàng khớp sát với các điểm dữ liệu. Tuy nhiên, khi kích thước tập huấn luyện tăng lên, giá trị  $R^2$  giảm dần và ổn định ở mức khoảng 0.620. Xu hướng này là bình thường, vì khi có nhiều dữ liệu hơn, mô hình phải khái quát hóa để bao quát tốt hơn các mẫu dữ liệu đa dạng hơn, thay vì khớp chặt chẽ với các điểm cụ thể trong tập nhỏ.

Sự giảm nhẹ và ổn định này của  $R^2$  cho thấy mô hình đang chuyển từ việc khớp với các điểm riêng lẻ sang khái quát hóa dự đoán cho tập dữ liệu

lớn hơn, giúp đảm bảo độ chính xác và tính ứng dụng cao hơn khi đối mặt với dữ liệu mới trong thực tế.

### **Đường kiểm tra (Test):**

Khi bắt đầu, hiệu suất của mô hình trên tập kiểm tra khá thấp (dưới 0.610), điều này có thể là do mô hình chưa được huấn luyện đủ để nhận diện các đặc trưng quan trọng từ dữ liệu kiểm tra, đặc biệt khi kích thước tập huấn luyện còn nhỏ. Tuy nhiên, khi kích thước của tập huấn luyện tăng lên, mô hình có cơ hội học từ nhiều mẫu dữ liệu hơn và có thể khái quát hóa tốt hơn, dẫn đến **hiệu suất trên tập kiểm tra cải thiện** dần dần. Cuối cùng, giá trị này ổn định xung quanh **0.615**, cho thấy mô hình đã học được đủ thông tin để dự đoán chính xác hơn.

Điều này cũng cho thấy rằng, khi dữ liệu huấn luyện đủ lớn và đa dạng, mô hình có khả năng **khái quát hóa tốt hơn**, tức là khả năng áp dụng các kiến thức học được từ dữ liệu huấn luyện vào dự đoán trên các dữ liệu mới (tập kiểm tra). Đây là một xu hướng tích cực, vì mô hình không chỉ phù hợp với dữ liệu huấn luyện mà còn có thể tạo ra các dự đoán chính xác và đáng tin cậy khi áp dụng vào thực tế.

### **Đánh giá overfitting:**

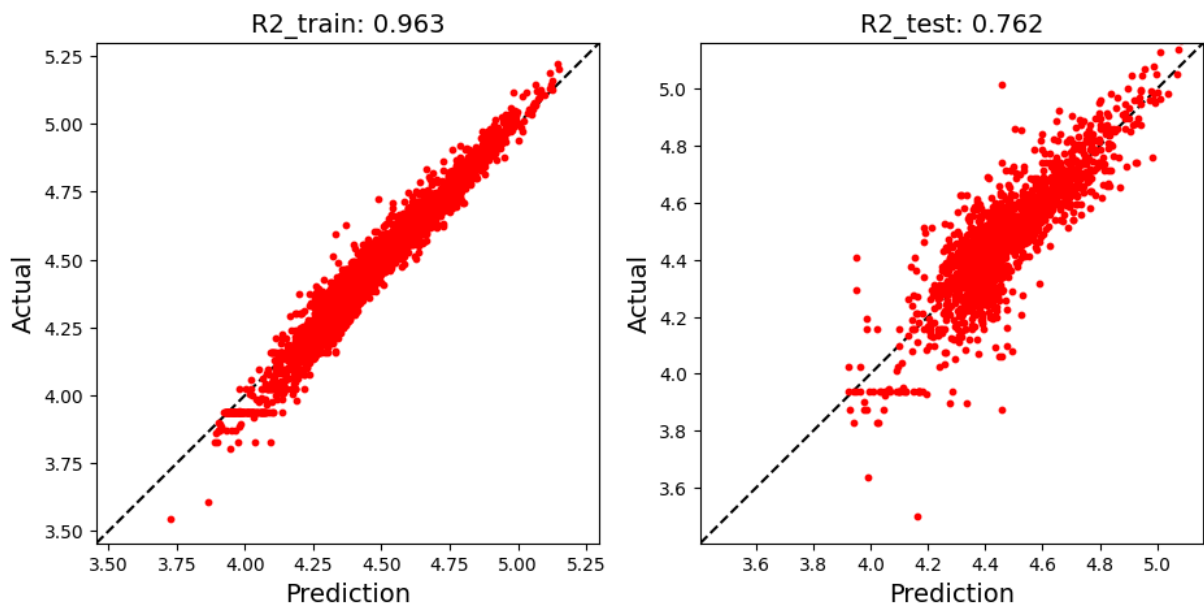
Khoảng cách giữa hai đường biểu diễn độ chính xác trên tập huấn luyện và tập kiểm tra là **không quá lớn**, và **khi kích thước tập huấn luyện tăng lên**, hai đường này dần **hội tụ lại với nhau**. Điều này cho thấy rằng mô hình có thể khái quát hóa tốt hơn khi có nhiều dữ liệu huấn luyện, giúp giảm thiểu hiện tượng overfitting.

### **Mức độ ổn định:**

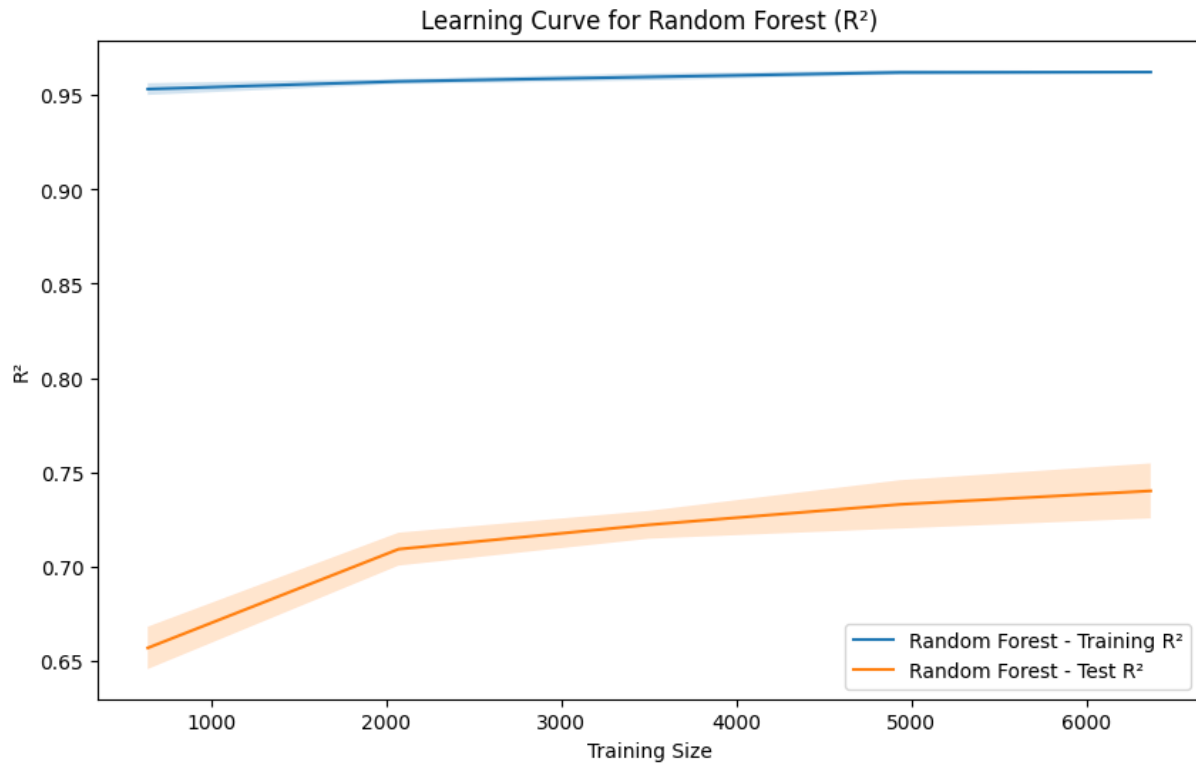
Khi kích thước tập huấn luyện đủ lớn, **cả hai đường (Training và Test)** đều trở nên **ổn định**, cho thấy mô hình đã đạt được mức độ **khái quát**

**hóa tốt.** Điều này có nghĩa là mô hình không chỉ học tốt từ dữ liệu huấn luyện mà còn có khả năng **dự đoán chính xác trên dữ liệu chưa thấy**, mà không bị quá khớp với các mẫu huấn luyện cụ thể. **Không có dấu hiệu overfitting nghiêm trọng**, và mô hình có khả năng làm việc hiệu quả trên cả dữ liệu huấn luyện và dữ liệu kiểm tra, giúp nó hoạt động tốt trong các tình huống thực tế.

- **Random Forest:**



Hình 1-18 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Random Forest



Hình 1-19 Learning Curve của Random Forest

#### **Đường huấn luyện (Training):**

Khi đường huấn luyện (Training) của mô hình Random Forest đạt mức rất cao, gần 0.95-0.96, và duy trì ổn định dù kích thước tập huấn luyện tăng, điều này chỉ ra rằng mô hình Random Forest có khả năng vừa khít rất tốt với dữ liệu huấn luyện. Mô hình có thể học và khớp chính xác với các điểm dữ liệu trong tập huấn luyện, đặc biệt là với các đặc trưng phức tạp hoặc phi tuyến tính của dữ liệu.

Tuy nhiên, khi thấy độ chính xác huấn luyện rất cao, cũng cần kiểm tra kết quả trên tập kiểm tra để chắc chắn rằng mô hình không chỉ học thuộc dữ liệu huấn luyện mà còn có thể áp dụng tốt trên dữ liệu chưa thấy, tức là có khả năng khái quát hóa tốt cho các tình huống thực tế.

#### **Đường kiểm tra (Test):**

Ban đầu, độ chính xác trên tập kiểm tra thấp hơn nhiều so với tập huấn luyện (khoảng 0.65). Tuy nhiên, khi kích thước tập huấn luyện tăng, độ chính xác trên tập kiểm tra dần cải thiện và ổn định quanh mức 0.75 khi tập huấn luyện đủ lớn. Điều này cho thấy mô hình cải thiện khả năng khái quát hóa khi được huấn luyện với nhiều dữ liệu hơn, mặc dù hiệu suất trên tập kiểm tra vẫn thấp hơn so với tập huấn luyện, phản ánh sự khác biệt giữa việc mô hình khớp tốt với dữ liệu huấn luyện và khả năng dự đoán trên dữ liệu chưa thấy.

#### **Đánh giá overfitting:**

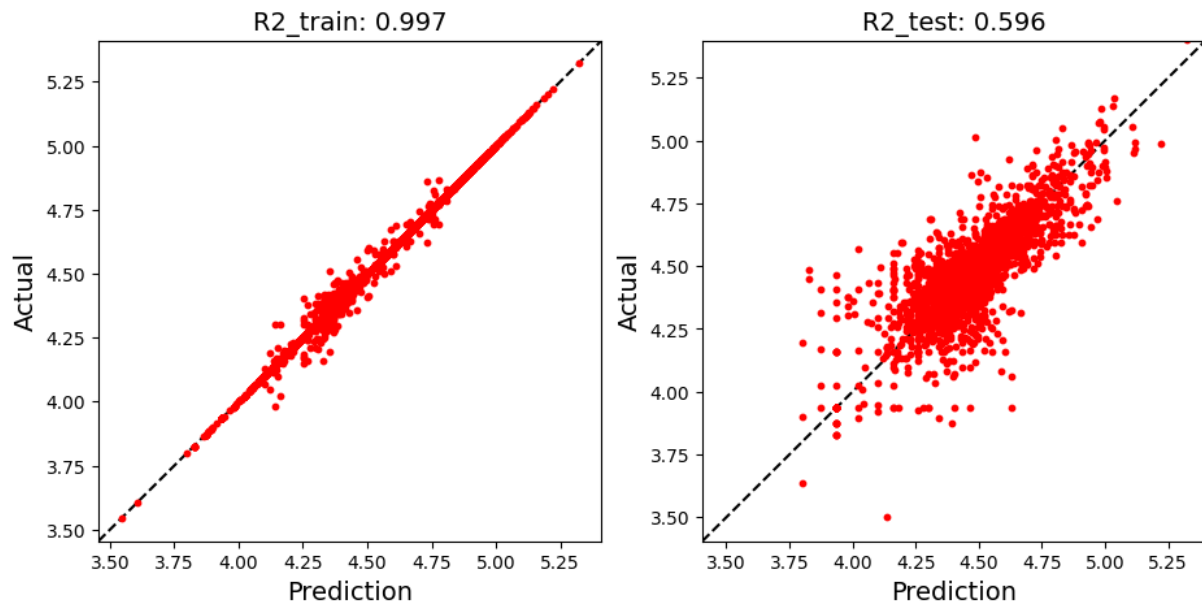
Khoảng cách lớn giữa  $R^2$  của tập huấn luyện (rất cao) và  $R^2$  của tập kiểm tra (khoảng 0.75) là dấu hiệu của **overfitting**. Mô hình học rất tốt trên dữ liệu huấn luyện, nhưng không thể **khái quát hóa tốt** trên dữ liệu kiểm tra. Điều này có nghĩa là mô hình quá khớp với dữ liệu huấn luyện và không thể dự đoán chính xác trên các dữ liệu mới, chưa thấy.

#### **Mức độ ổn định:**

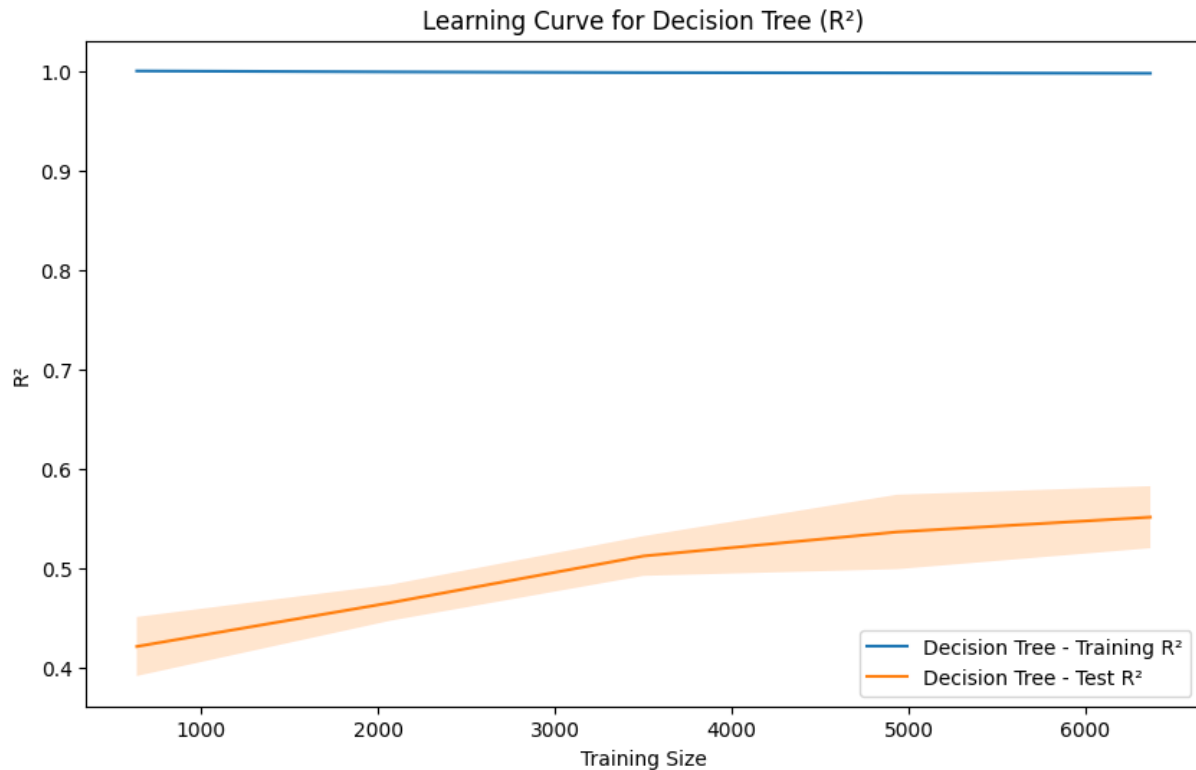
Dù đường Test có xu hướng ổn định khi kích thước tập huấn luyện tăng, nhưng nó vẫn không tiến gần đến mức của Training, điều này cho thấy mô hình vẫn gặp khó khăn trong việc khái quát hóa và hiệu suất trên tập kiểm tra vẫn thấp hơn so với tập huấn luyện. Sự khác biệt này là một dấu hiệu rõ rệt của overfitting.



- **Decision Tree:**



Hình 1-20 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Decision Tree:



Hình 1-21 Learning Curve của Decision Tree

### **Đường huấn luyện (Training):**

Khi đường huấn luyện duy trì ở mức  $R^2 = 1$ , điều này có nghĩa là mô hình hoàn toàn phù hợp với dữ liệu huấn luyện, tức là dự đoán của mô hình trùng khớp chính xác với giá trị thực tế trong tập huấn luyện. Đây là một dấu hiệu rõ ràng của hiện tượng overfitting.

### **Đường kiểm tra (Test):**

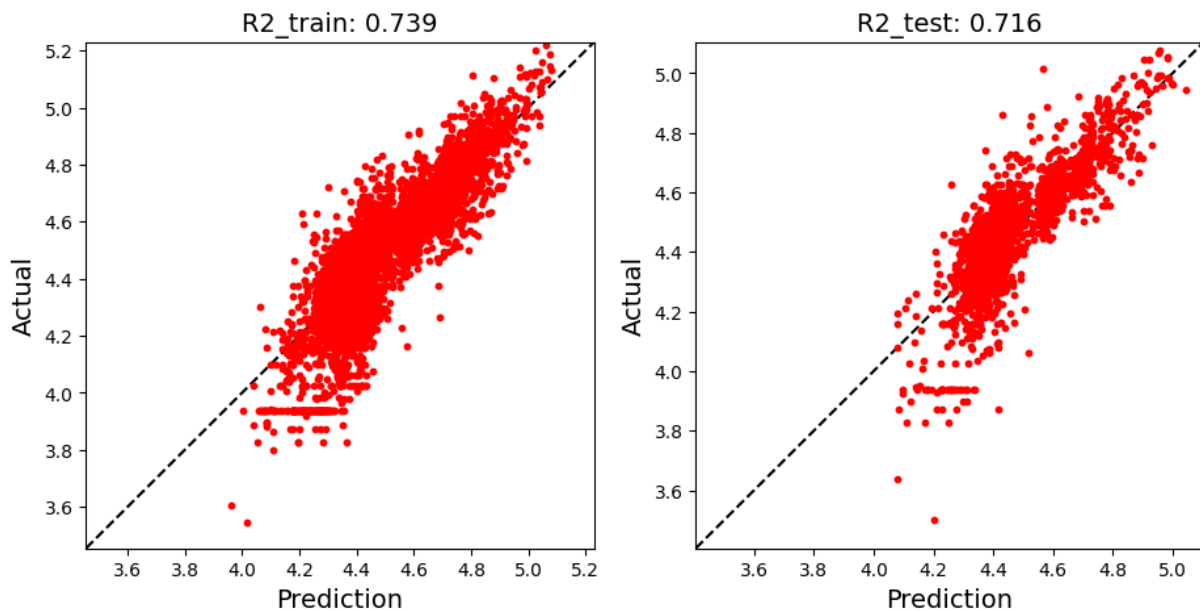
Khi đường của tập kiểm tra tăng chậm theo kích thước tập huấn luyện và chỉ đạt  $R^2 \approx 0.5$  ở phần cuối cùng, điều này cho thấy mô hình hoạt động kém trên tập kiểm tra. Mặc dù mô hình có thể học được các đặc trưng từ dữ liệu huấn luyện, nhưng khả năng tổng quát hóa kém cho thấy nó không thể áp dụng hiệu quả cho các dữ liệu mới, chưa thấy trước.

$R^2 = 0.5$  trên tập kiểm tra cho thấy mô hình chỉ giải thích được một nửa sự biến động của dữ liệu, điều này là không đủ để đưa ra những dự đoán chính xác trong thực tế. Việc tăng chậm này phản ánh rằng mô hình cần một lượng dữ liệu lớn hơn hoặc các điều chỉnh tốt hơn để học được các mối quan hệ tổng quát hơn trong dữ liệu và giảm sự phụ thuộc vào các mẫu huấn luyện cụ thể.

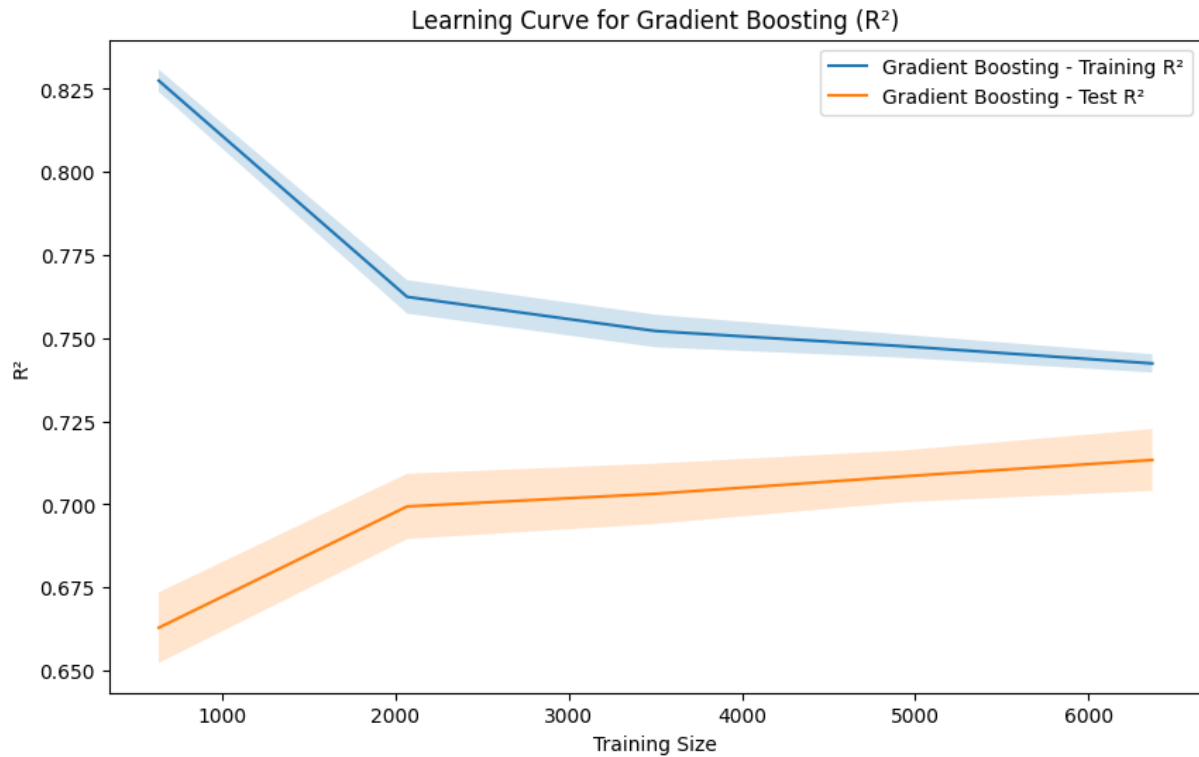
#### Đánh giá overfitting:

Mô hình **Decision Tree** trong trường hợp này có dấu hiệu **overfitting** rõ ràng. Dù mô hình đạt được **độ chính xác hoàn hảo trên tập huấn luyện** với  $R^2 = 1$ , nhưng khi áp dụng vào **tập kiểm tra**, hiệu suất lại **rất kém**, chỉ đạt khoảng  $R^2 = 0.5$ . Điều này cho thấy mô hình chỉ học được rất chi tiết các mẫu trong tập huấn luyện mà không thể **khái quát hóa tốt** cho các dữ liệu chưa thấy.

- **Gradient Boosting:**



Hình 1-22 Hình ảnh mô tả giá trị dự đoán và thực tế trên tập train và test của Gradient Boosting



Hình 1-23 Learning Curve của Gradient Boosting

### Đường huấn luyện (Training):

Mức ổn định của  $R^2$  quanh  $\sim 0.75$  khi kích thước tập huấn luyện đủ lớn cho thấy rằng mô hình đã đạt được sự khái quát hóa tốt và không bị overfitting, vì sự giảm dần này là một tín hiệu cho thấy mô hình không chỉ học các đặc điểm cụ thể của tập huấn luyện mà còn có thể dự đoán chính xác hơn trên dữ liệu chưa thấy. Điều này chỉ ra rằng mô hình có thể áp dụng được trong thực tế và có khả năng giải thích các mẫu dữ liệu mới.

### Đường kiểm tra (Test):

$R^2$  trên tập kiểm tra **ổn định quanh mức  $\sim 0.7$**  khi kích thước tập huấn luyện lớn. Điều này cho thấy mô hình đang có khả năng **dự đoán chính xác** trên **dữ liệu chưa thấy trước** mà không quá phụ thuộc vào các đặc trưng

riêng biệt trong tập huấn luyện. Mặc dù vẫn có một khoảng cách giữa  **$R^2$  của tập huấn luyện và tập kiểm tra**, nhưng sự ổn định của đường kiểm tra ở mức  $\sim 0.7$  chỉ ra rằng mô hình **khái quát hóa tốt** và có thể áp dụng vào các tình huống thực tế với hiệu suất khá ổn định.

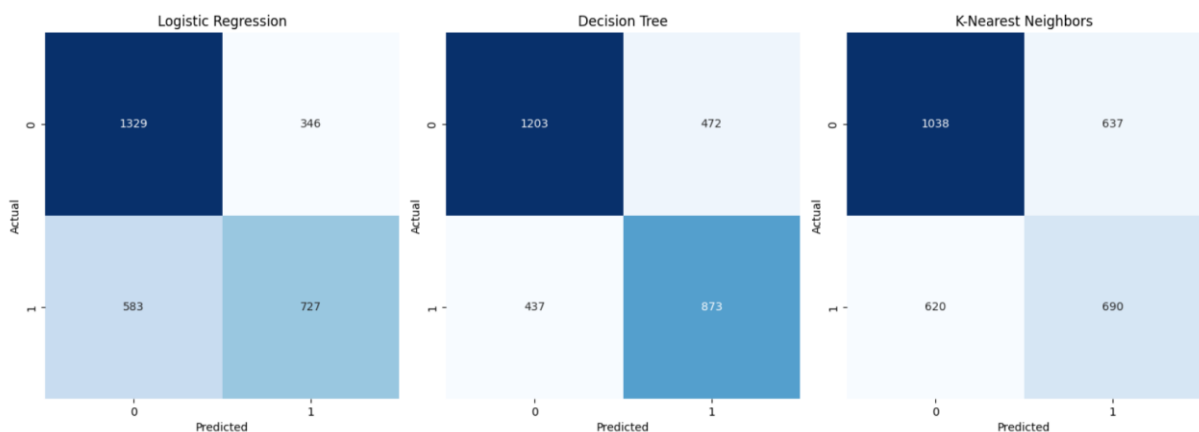
#### Đánh giá overfitting:

Trong mô hình **Gradient Boosting**, không có dấu hiệu rõ ràng của **overfitting**. Khoảng cách giữa **Training  $R^2$**  và **Test  $R^2$**  là **nhỏ và ổn định**, cho thấy mô hình có **khả năng tổng quát hóa tốt** và không bị quá khớp với dữ liệu huấn luyện. Điều này chỉ ra rằng mô hình có thể **dự đoán chính xác** cả trên dữ liệu huấn luyện và dữ liệu kiểm tra, giúp nó **áp dụng tốt hơn** vào các tình huống thực tế.

Mặc dù mô hình có hiệu suất khá ổn định, khoảng cách giữa **hai đường này** có thể được cải thiện thêm. **Tinh chỉnh siêu tham số** hoặc **thêm nhiều dữ liệu huấn luyện** có thể giúp mô hình **tối ưu hóa hiệu suất** hơn nữa, làm cho khả năng tổng quát hóa của nó trở nên mạnh mẽ hơn, giảm bớt sự chênh lệch giữa dữ liệu huấn luyện và kiểm tra.

### 1.4.3 Phân loại

#### 1.4.3.1 Đánh giá qua ma trận nhầm lẫn



Hình 1-24 Confusion Matrix

### **Logistic Regression:**

- True Negatives (TN): 1329 — số lượng mẫu mà mô hình dự đoán chính xác là 0 (âm tính) khi thực tế cũng là 0.
- False Positives (FP): 346 — số lượng mẫu mà mô hình dự đoán là 1 (dương tính) khi thực tế là 0.
- False Negatives (FN): 583 — số lượng mẫu mà mô hình dự đoán là 0 khi thực tế là 1.
- True Positives (TP): 727 — số lượng mẫu mà mô hình dự đoán chính xác là 1 khi thực tế cũng là 1.

=> Logistic Regression có tỷ lệ dự đoán chính xác tương đối cao cho các mẫu thuộc lớp 0, nhưng tỷ lệ dự đoán đúng cho lớp 1 còn thấp. Mô hình này dường như nghiêng về dự đoán lớp 0 hơn.

### **Decision Tree**

- True Negatives (TN): 1203 — số lượng mẫu mà mô hình dự đoán chính xác là 0 khi thực tế cũng là 0.
- False Positives (FP): 472 — số lượng mẫu mà mô hình dự đoán là 1 khi thực tế là 0.
- False Negatives (FN): 437 — số lượng mẫu mà mô hình dự đoán là 0 khi thực tế là 1.
- True Positives (TP): 873 — số lượng mẫu mà mô hình dự đoán chính xác là 1 khi thực tế cũng là 1.

=> Decision Tree có khả năng cân bằng hơn giữa hai lớp 0 và 1 so với Logistic Regression. Số lượng TP cao hơn và FN thấp hơn so với Logistic Regression, cho thấy mô hình này có khả năng dự đoán tốt hơn cho lớp 1, mặc dù FP tăng cao hơn.

### **K-Nearest Neighbors (KNN)**

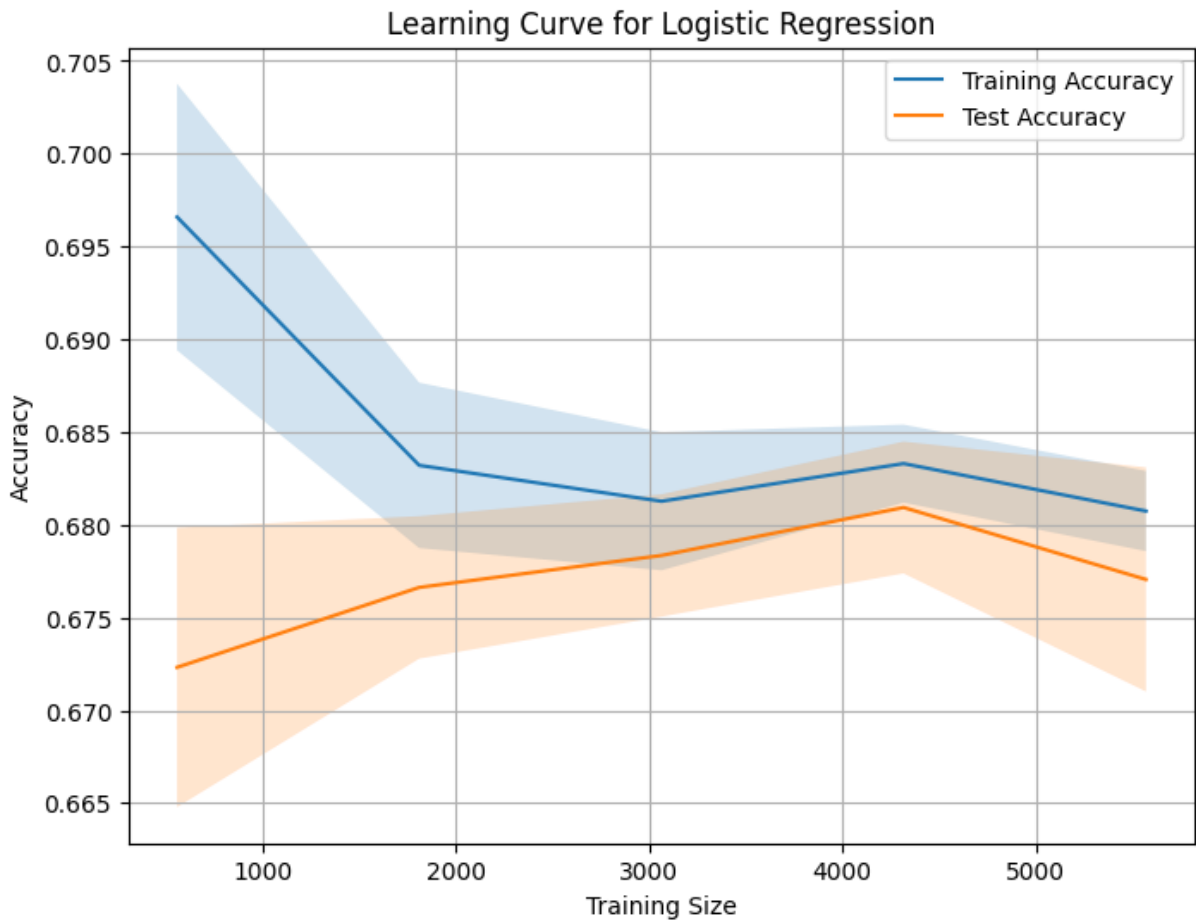
- True Negatives (TN): 1038 — số lượng mẫu mà mô hình dự đoán chính xác là 0 khi thực tế cũng là 0.
- False Positives (FP): 637 — số lượng mẫu mà mô hình dự đoán là 1 khi thực tế là 0.
- False Negatives (FN): 620 — số lượng mẫu mà mô hình dự đoán là 0 khi thực tế là 1.
- True Positives (TP): 690 — số lượng mẫu mà mô hình dự đoán chính xác là 1 khi thực tế cũng là 1.

=>KNN có tỷ lệ dự đoán đúng thấp hơn ở cả hai lớp so với Logistic Regression và Decision Tree. Số lượng FN và FP đều cao hơn, cho thấy mô hình này khó khăn trong việc phân biệt giữa hai lớp trong tập dữ liệu này.

**Kết luận:** Nhìn chung, Decision Tree có vẻ là lựa chọn tốt nhất trong số ba mô hình này do khả năng cân bằng và dự đoán chính xác cao hơn cho cả hai lớp.

### 1.4.3.2 Đánh giá qua Learning Curve

#### Logistic Regression:



Hình 1-25 Learning Curve của Logistic Regression

**Tập huấn luyện (Training):** Đường accuracy của tập huấn luyện bắt đầu cao (~0.69) nhưng giảm nhẹ khi kích thước tập huấn luyện tăng và ổn định ở mức ~0.68. Điều này cho thấy mô hình học tốt từ dữ liệu huấn luyện nhưng có sự khái quát hóa khi dữ liệu huấn luyện tăng lên.

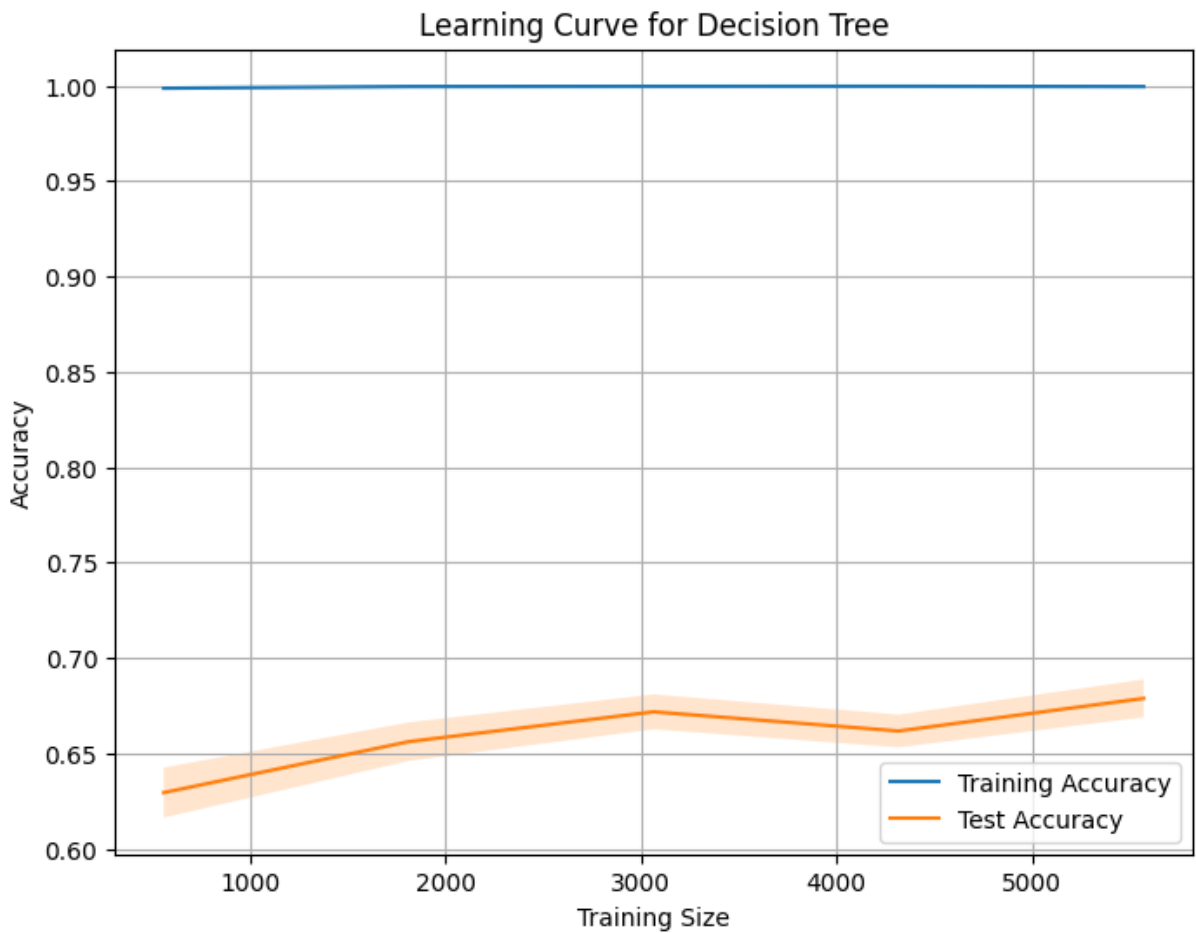
**Tập kiểm tra (Test):** Đường accuracy của tập kiểm tra bắt đầu ở mức thấp (~0.67) nhưng dần tăng lên khi kích thước tập huấn luyện lớn hơn và ổn định ở mức



~0.68. Điều này chứng tỏ mô hình cải thiện khả năng dự đoán trên dữ liệu kiểm tra khi có nhiều dữ liệu huấn luyện hơn.

**Kết hợp train và test:** Hai đường accuracy của tập huấn luyện và tập kiểm tra khá gần nhau và ổn định khi kích thước tập huấn luyện tăng. Khoảng cách giữa chúng không quá lớn, cho thấy mô hình không bị overfitting hay underfitting, và có khả năng tổng quát tốt trên dữ liệu chưa thấy.

### Decision Tree:



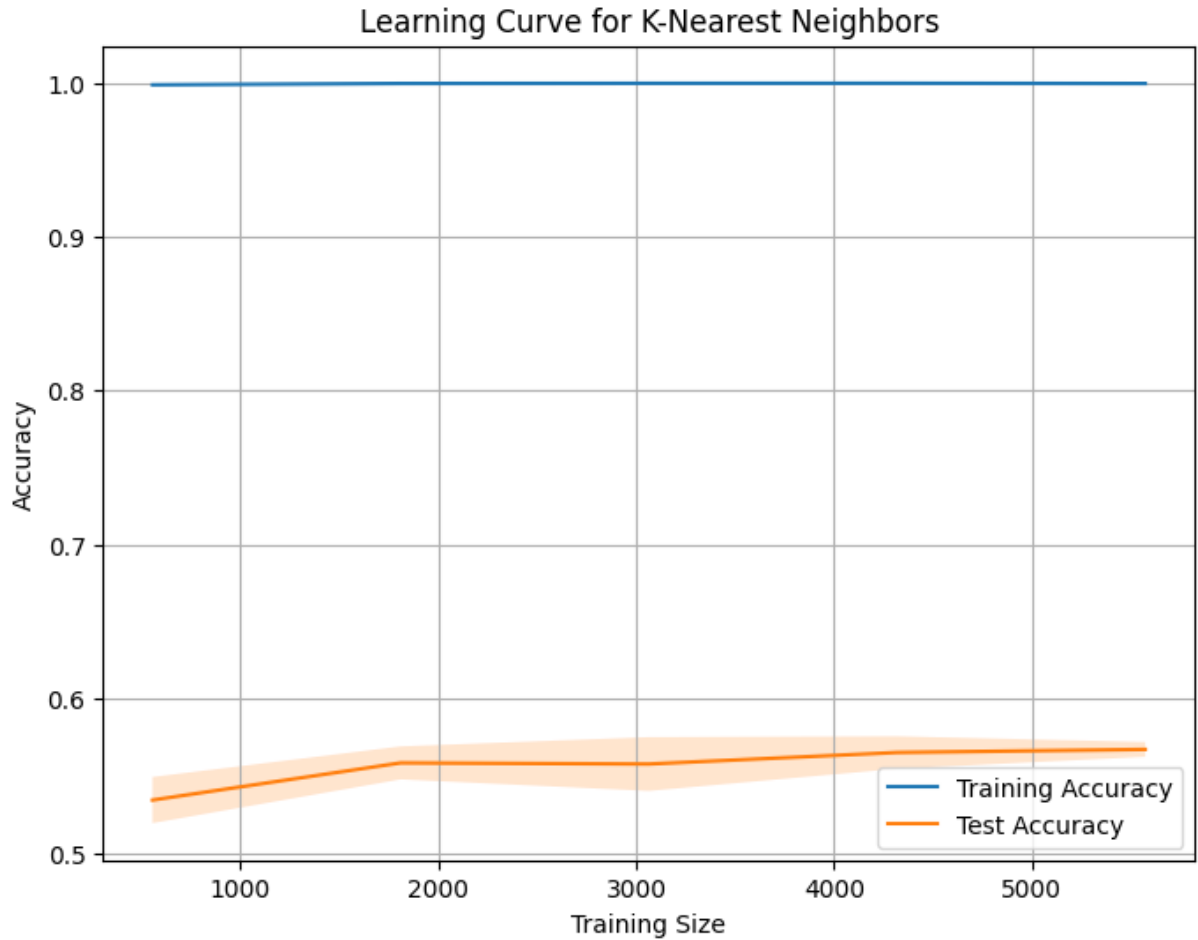
Hình 1-26 Learning Curve của Decision Tree

**Training Set:** Mô hình Decision Tree hoàn toàn khớp với dữ liệu huấn luyện, thể hiện qua accuracy rất cao, gần như đạt giá trị tối đa. Điều này cho thấy mô hình học rất chi tiết các mẫu trong tập huấn luyện, nhưng có nguy cơ overfitting.

**Test Set:** Accuracy của tập kiểm tra thấp hơn đáng kể, dao động trong khoảng 0.65-0.7, cho thấy mô hình không tổng quát tốt khi áp dụng vào dữ liệu chưa thấy trước.

**Kết hợp Training và Test:** Khoảng cách giữa accuracy của tập huấn luyện và tập kiểm tra là dấu hiệu rõ ràng của overfitting. Mô hình học quá chi tiết trên dữ liệu huấn luyện nhưng không khái quát tốt trên dữ liệu mới. Để cải thiện, có thể giảm độ sâu của cây quyết định hoặc áp dụng kỹ thuật regularization để tăng khả năng tổng quát và giảm overfitting.

### K-Nearest Neighbors (KNN)



Hình 1-27 Learning Curve của K-Nearest Neighbors (KNN)

**Training Set:** Đường accuracy của tập huấn luyện đạt mức tối đa 1.0, cho thấy mô hình KNN hoàn toàn khớp với dữ liệu huấn luyện, điều này thường dẫn đến khả năng overfitting.

**Test Set:** Đường accuracy của tập kiểm tra thấp hơn nhiều, dao động trong khoảng 0.53-0.58, cho thấy mô hình không tổng quát tốt khi áp dụng vào dữ liệu chưa thấy trước.

**Kết hợp Training và Test:** Khoảng cách rõ rệt giữa accuracy của tập huấn luyện và tập kiểm tra là dấu hiệu của overfitting. Mô hình học rất tốt trên dữ liệu huấn luyện nhưng không khái quát tốt trên dữ liệu kiểm tra. Để cải thiện, có thể áp dụng các kỹ thuật regularization hoặc giảm độ phức tạp của mô hình để cải thiện khả năng tổng quát.

## CHƯƠNG 2 - CÂU 2

### 2.1 Tổng quan overfitting

#### 2.1.1 Khái niệm overfitting

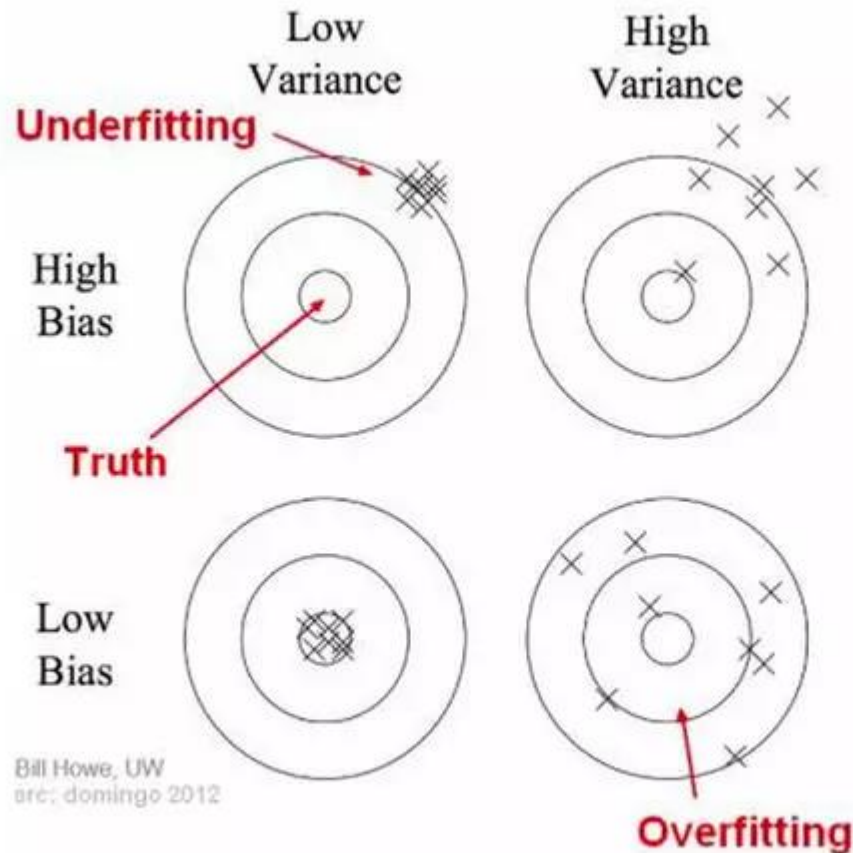
Overfitting là hiện tượng xảy ra khi một mô hình học quá chi tiết từ dữ liệu huấn luyện, dẫn đến việc mô hình không thể tổng quát hóa tốt cho các dữ liệu chưa thấy trước. Cụ thể, khi một mô hình bị overfit, nó sẽ "học" các đặc điểm không quan trọng hoặc nhiễu (noise) trong dữ liệu huấn luyện thay vì chỉ học các xu hướng chính. Hệ quả là, mặc dù mô hình có thể đạt độ chính xác rất cao trên tập huấn luyện, nhưng khi áp dụng vào tập kiểm tra hoặc dữ liệu thực tế, mô hình lại có hiệu suất kém, vì nó không thể dự đoán chính xác các mẫu dữ liệu mới.

Dấu hiệu dễ nhận biết của overfitting là khi độ chính xác trên tập huấn luyện rất cao, thậm chí gần như hoàn hảo, nhưng độ chính xác trên tập kiểm tra lại thấp hơn nhiều. Khoảng cách giữa độ chính xác của tập huấn luyện và tập kiểm tra thường phản ánh rõ rệt hiện tượng này. Nguyên nhân chủ yếu gây ra overfitting là do mô hình quá phức tạp, ví dụ như một cây quyết định có độ sâu quá lớn, hoặc khi dữ liệu huấn luyện không đủ đa dạng và chỉ bao gồm một tập hợp nhỏ các mẫu.

#### 2.1.2 Bias và variance

Bias là sự sai lệch giữa giá trị mà model chúng ta dự đoán được với giá trị thật (predict - ground truth). Mô hình với low bias thì chênh lệch giữa giá trị dự đoán và giá trị thật nhỏ  $\Rightarrow$  Mô hình tốt. Và ngược lại high bias thì chênh lệch giữa predict và ground truth lớn  $\Rightarrow$  Mô hình lỗi cao trên cả tập huấn luyện (training) và tập kiểm thử (testing)  $\Rightarrow$  Underfitting.

Variance đại diện cho độ phân tán dữ liệu. Variance cao chứng tỏ phân tán cao, tập trung chủ yếu vào dữ liệu huấn luyện mà không mang được tính tổng quát trên dữ liệu chưa gặp bao giờ => Mô hình rất tốt trên tập dữ liệu huấn luyện nhưng kết quả rất tệ trên tập kiểm thử => Overfitting.



Hình 2-1 Mô tả bias và variance

### 2.1.3 Lý do overfitting

**Mô hình quá phức tạp:** Khi mô hình có quá nhiều tham số hoặc độ phức tạp cao, chẳng hạn như độ sâu lớn trong cây quyết định hoặc số lượng tầng dày đặc trong mạng nơ-ron, nó có xu hướng "ghi nhớ" dữ liệu huấn luyện thay

vì khái quát hóa. Những mô hình phức tạp có khả năng học các chi tiết rất nhỏ, kể cả nhiễu trong dữ liệu.

**Dữ liệu huấn luyện không đủ lớn hoặc không đa dạng:** Nếu dữ liệu huấn luyện quá nhỏ hoặc không đa dạng, mô hình sẽ khó học được các xu hướng chính và dễ dẫn đến việc học quá chi tiết các đặc điểm của tập huấn luyện. Điều này làm giảm khả năng tổng quát hóa trên dữ liệu mới.

**Dữ liệu huấn luyện có nhiễu nhiều (noise):** Nếu dữ liệu huấn luyện chứa nhiều điểm dữ liệu ngoại lai hoặc các đặc điểm không liên quan, mô hình có thể "học" cả các nhiễu này, khiến hiệu suất trên dữ liệu mới giảm sút.

**Không áp dụng regularization (chuẩn hóa):** Regularization là kỹ thuật điều chỉnh để giảm độ phức tạp của mô hình và ngăn chặn overfitting. Nếu không áp dụng regularization, mô hình sẽ không được kiểm soát độ phức tạp và dễ học quá mức từ dữ liệu huấn luyện.

**Thực hiện quá nhiều lần huấn luyện:** Khi điều chỉnh mô hình quá mức trên dữ liệu huấn luyện (chẳng hạn như huấn luyện quá nhiều vòng), mô hình sẽ tiếp tục "nhớ" các mẫu trong dữ liệu mà không học được cách tổng quát hóa. Điều này dễ xảy ra với các mô hình có khả năng ghi nhớ cao như mạng nơ-ron.

**Thiếu phương pháp đánh giá chính xác:** Nếu mô hình không được đánh giá kỹ trên dữ liệu kiểm tra hoặc không sử dụng các kỹ thuật đánh giá như cross-validation, khả năng overfitting sẽ cao hơn. Việc không có đánh giá chính xác làm tăng rủi ro tối ưu hóa quá mức cho dữ liệu huấn luyện.

**Chưa xử lý trước dữ liệu hợp lý:** Các đặc điểm không liên quan hoặc có mối quan hệ chông chéo trong dữ liệu cũng dễ dẫn đến overfitting. Vì vậy, nếu chưa thực hiện các bước xử lý trước như loại bỏ các đặc trưng dư thừa hoặc chuẩn hóa dữ liệu, mô hình sẽ học các đặc điểm không cần thiết.

#### 2.1.4 Biện pháp

**Sử dụng Regularization:** Thêm các thuật toán regularization như L1 và L2 vào hàm mất mát của mô hình sẽ giúp giảm thiểu độ phức tạp của mô hình, ngăn cản mô hình học các mẫu quá chi tiết. L1 regularization có thể làm cho một số trọng số bằng 0, trong khi L2 sẽ giữ các trọng số nhỏ hơn, cả hai đều giúp giảm overfitting.

**Thu thập thêm dữ liệu huấn luyện:** Nếu có thể, tăng kích thước và độ đa dạng của dữ liệu huấn luyện sẽ giúp mô hình học được các xu hướng chính và tăng khả năng khái quát hóa. Thêm dữ liệu mới từ nhiều nguồn khác nhau hoặc thực hiện các kỹ thuật tăng cường dữ liệu (data augmentation) có thể mang lại hiệu quả.

**Sử dụng kỹ thuật cross-validation:** Cross-validation, đặc biệt là kỹ thuật k-fold cross-validation, giúp kiểm tra mô hình trên nhiều tập dữ liệu khác nhau, từ đó tăng tính khách quan và ngăn chặn việc tối ưu hóa quá mức chỉ cho một tập dữ liệu nhất định. Cross-validation cũng giúp lựa chọn mô hình và tinh chỉnh siêu tham số tốt hơn.

**Giảm độ phức tạp của mô hình:** Nếu mô hình quá phức tạp, bạn có thể giảm số lượng tham số, tầng hoặc độ sâu của mô hình (như giảm số nút trong cây quyết định hoặc số tầng trong mạng nơ-ron). Việc giảm độ phức tạp sẽ làm mô hình bớt tập trung vào các chi tiết nhỏ, giúp khái quát hóa tốt hơn.

**Sử dụng Dropout trong mạng nơ-ron:** Dropout là một kỹ thuật được sử dụng trong mạng nơ-ron, trong đó một số nơ-ron sẽ được "tắt" ngẫu nhiên trong quá trình huấn luyện. Dropout ngăn mô hình quá phụ thuộc vào một số nơ-ron cụ thể, giúp phân bố trọng số và giảm overfitting.

**Sử dụng kỹ thuật Early Stopping:** Early stopping là kỹ thuật dừng quá trình huấn luyện khi hiệu suất trên tập kiểm tra ngừng cải thiện hoặc bắt đầu



giảm. Điều này ngăn mô hình huấn luyện quá mức, từ đó tránh hiện tượng overfitting.

**Chuẩn hóa dữ liệu (Data Normalization):** Chuẩn hóa dữ liệu giúp tránh trường hợp các đặc trưng có quy mô khác nhau gây ra sự ưu tiên không mong muốn. Chuẩn hóa dữ liệu có thể làm giảm sự phức tạp và nhiễu trong dữ liệu, từ đó giúp mô hình học tập hiệu quả hơn.

**Giảm số lượng đặc trưng (Feature Selection):** Loại bỏ các đặc trưng không liên quan hoặc có tương quan cao với nhau có thể giảm thiểu overfitting. Điều này giúp mô hình tập trung vào các đặc trưng quan trọng và tránh bị phân tán vào các đặc trưng không cần thiết.

**Sử dụng Ensemble Methods:** Các phương pháp ensemble như Bagging và Boosting (ví dụ, Random Forest và Gradient Boosting) kết hợp dự đoán từ nhiều mô hình khác nhau để tăng cường độ chính xác và giảm overfitting. Ensemble methods thường tạo ra các mô hình ổn định và khái quát hóa tốt hơn.

**Điều chỉnh siêu tham số (Hyperparameter Tuning):** Sử dụng các kỹ thuật như Grid Search hoặc Random Search để tìm các siêu tham số tối ưu cho mô hình. Điều này giúp tìm ra cấu hình tối ưu cho mô hình, ngăn chặn khả năng quá tải và cải thiện khả năng khái quát hóa.

## 2.2 Sử dụng hyperparameter tuning, cross-validation, regularization vào model

### 2.2.1 Hồi quy

#### Random Forest Regressor:

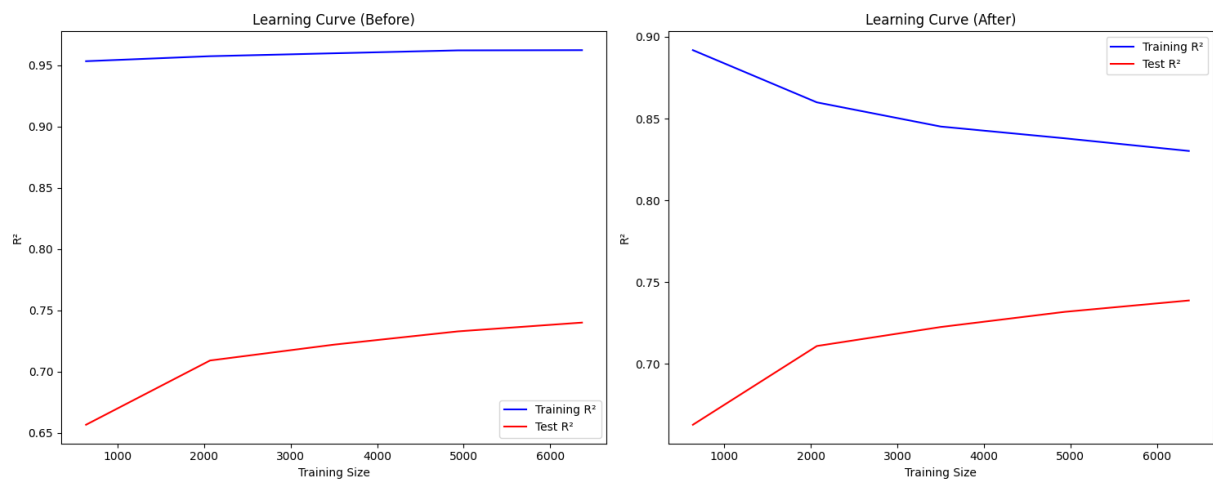
Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

```
'bootstrap': True,
'max_depth': 10,
```

```
'min_samples_leaf': 2,
'min_samples_split': 2,
'n_estimators': 200,
'ccp_alpha': 0.0
```

	Metric	Trước overfitting	Sau overfitting
0	MAE	0.067292	0.069487
1	RMSE	0.094621	0.096317
2	R <sup>2</sup>	0.761969	0.753357

Hình 2-2 Các tham số trước và sau xử lý của Random Forest Regressor



Hình 2-3 Learning Curve trước và sau điều chỉnh của Random Forest Regressor

Trước khi xử lý overfitting, có thể thấy qua đường cong học tập và các chỉ số hiệu suất rằng mô hình đang quá tập trung vào dữ liệu huấn luyện, nhưng lại không áp dụng tốt trên dữ liệu kiểm tra. Cụ thể:

### **Đường cong học tập của $R^2$ :**

$R^2$  của tập huấn luyện cao (khoảng 0.90), chứng tỏ mô hình đạt hiệu suất tốt trên dữ liệu huấn luyện. Tuy nhiên, sự chênh lệch lớn so với tập kiểm tra báo hiệu hiện tượng overfitting.

$R^2$  của tập kiểm tra thấp hơn và tăng dần khi kích thước dữ liệu huấn luyện tăng, nhưng vẫn duy trì khoảng cách lớn với  $R^2$  của tập huấn luyện, cho thấy mô hình chưa thể tổng quát hóa tốt trên dữ liệu mới.

### **Chỉ số hiệu suất:**

Các chỉ số lỗi như MAE và RMSE thấp trên tập huấn luyện, nhưng tăng cao trên tập kiểm tra, cho thấy mô hình có độ chính xác thấp trên dữ liệu kiểm tra. Điều này cũng được xác nhận bởi giá trị  $R^2$  thấp trên tập kiểm tra, biểu thị xu hướng overfit.

**Sau khi xử lý Overfitting**, mô hình đã có những cải thiện rõ rệt về khả năng tổng quát hóa, thể hiện qua các thay đổi sau:

### **Đường cong học tập của $R^2$ :**

$R^2$  trên tập huấn luyện giảm nhẹ, nhưng vẫn ở mức cao, cho thấy mô hình không còn quá tối ưu hóa trên dữ liệu huấn luyện.

$R^2$  trên tập kiểm tra tăng lên, và khoảng cách giữa  $R^2$  của hai tập (train và test) thu hẹp, chứng tỏ mô hình đã học cách tổng quát hóa và có thể áp dụng tốt hơn trên dữ liệu mới.

### **Chỉ số hiệu suất:**

MAE và RMSE trên tập kiểm tra giảm, cho thấy mô hình đã cải thiện độ chính xác trên dữ liệu mới. Trong khi đó, giá trị MAE và RMSE trên tập huấn luyện có thể tăng nhẹ, điều này là bình thường khi xử lý overfitting.

$R^2$  của tập kiểm tra cũng tăng lên, cho thấy mô hình hiện tại có khả năng giải thích tốt hơn biến động của dữ liệu kiểm tra, đạt được mục tiêu tổng quát hóa

### Decision Tree Regressor:

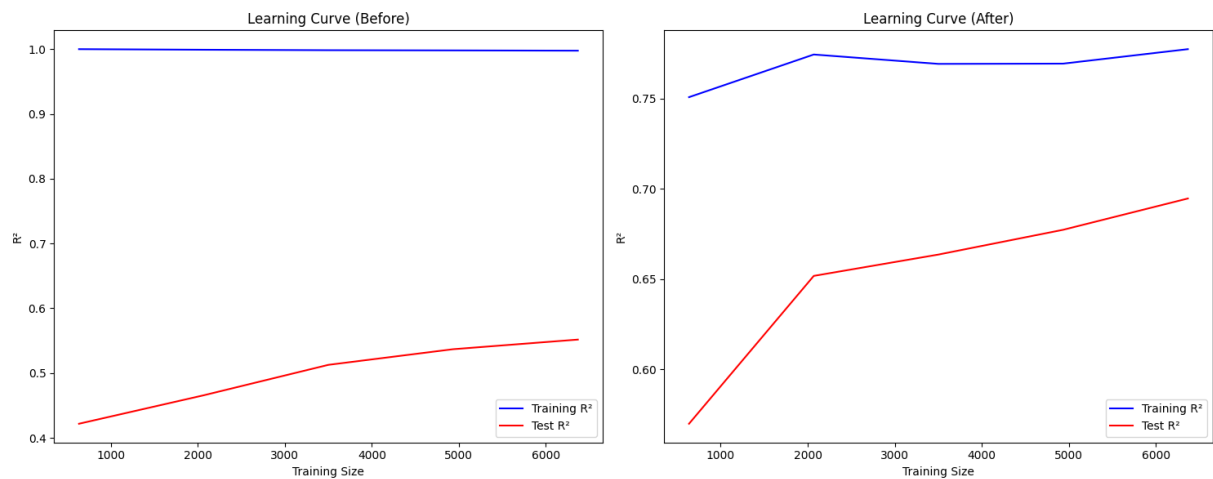
Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

```
'max_depth':10,
'min_samples_split':2,
'min_samples_leaf':10,
'max_features': None
```

**Metric Trước overfitting Sau overfitting**

0	MAE	0.084822	0.072752
1	RMSE	0.123314	0.102990
2	R <sup>2</sup>	0.595718	0.717997

Hình 2-4 Các tham số trước và sau xử lý của Decision Tree Regressor



Hình 2-5 Learning Curve trước và sau điều chỉnh của Decision Tree Regressor

**Trước khi xử lý overfitting, mô hình có các dấu hiệu quá khớp rõ ràng. Cụ thể:**

$R^2$  của tập huấn luyện cao hơn nhiều so với  $R^2$  của tập kiểm tra, điều này cho thấy mô hình đã học quá chi tiết các mẫu của tập huấn luyện, gây ra hiện tượng overfitting và làm giảm khả năng dự đoán trên dữ liệu mới.

Chỉ số lỗi MAE và RMSE cao trên tập kiểm tra, biểu hiện hiệu suất kém khi mô hình đối mặt với dữ liệu chưa thấy trước.

**Sau khi thực hiện các biện pháp giảm overfitting, mô hình đã đạt độ tổng quát tốt hơn:**

$R^2$  của tập huấn luyện giảm và tiến gần đến  $R^2$  của tập kiểm tra, cho thấy mô hình không còn phụ thuộc quá nhiều vào dữ liệu huấn luyện, giúp tăng khả năng áp dụng trên dữ liệu mới.

Chỉ số MAE và RMSE giảm trên tập kiểm tra, chứng minh rằng mô hình hiện tại có độ chính xác cao hơn khi dự đoán dữ liệu chưa được huấn luyện.

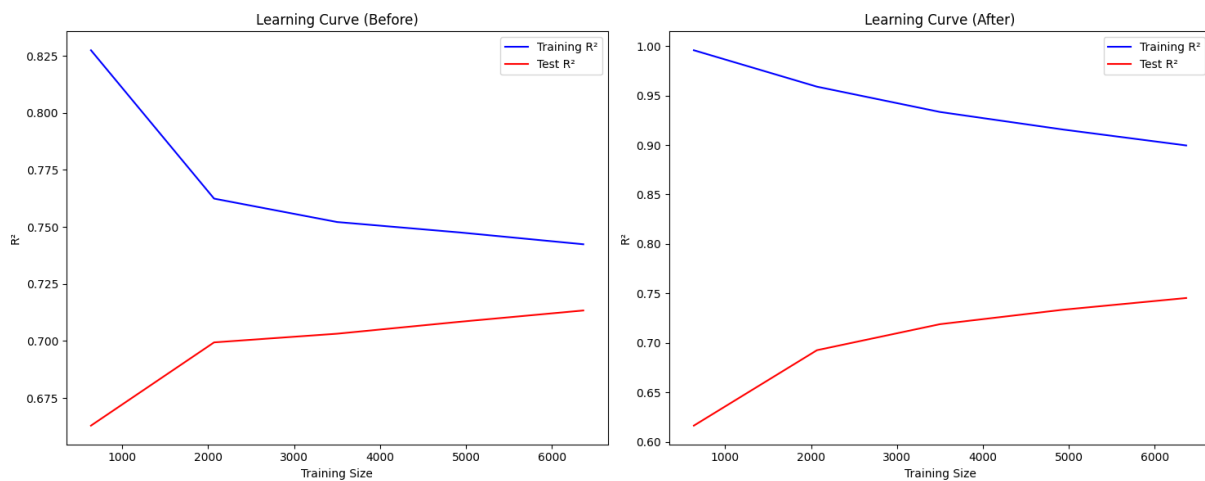
### **Gradient Boosting Regressor**

Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

```
'n_estimators':200,
'learning_rate':0.2,
'max_depth':5,
'min_samples_split':5,
'min_samples_leaf':2,
'subsample': 0.8
```

	Metric	Trước overfitting	Sau overfitting
0	MAE	0.073920	0.067693
1	RMSE	0.103331	0.093876
2	$R^2$	0.716130	0.765703

Hình 2-6 Các tham số trước và sau xử lý của Gradient Boosting Regressor



Hình 2-7 Learning Curve trước và sau điều chỉnh của Gradient Boosting Regressor

#### Trước khi xử lý overfitting:

- **Đường cong học tập của  $R^2$ :**
  - **$R^2$  trên tập huấn luyện rất cao (gần 1)**, cho thấy mô hình đạt hiệu suất tốt trên dữ liệu huấn luyện. Tuy nhiên, độ chính xác cao này có thể là dấu hiệu của overfitting, khi mô hình học cả nhiễu và chi tiết không cần thiết của dữ liệu huấn luyện.
  - **$R^2$  trên tập kiểm tra bắt đầu thấp (khoảng 0.6)** nhưng tăng dần khi kích thước tập huấn luyện lớn lên. Khoảng cách lớn giữa  $R^2$  của tập huấn luyện và tập kiểm tra cho thấy mô hình khó tổng quát hóa, tức là nó ghi

nhớ dữ liệu huấn luyện hơn là học các mẫu thực sự, dẫn đến hiệu suất kém trên dữ liệu mới.

- **Chỉ số hiệu suất:**

- **MAE và RMSE** đều thấp trên tập huấn luyện nhưng tăng lên trên dữ liệu kiểm tra, cho thấy độ chính xác kém khi gặp dữ liệu mới.
- Giá trị  **$R^2$  trên tập kiểm tra thấp hơn** so với tập huấn luyện, khẳng định rằng mô hình đang overfit và khó áp dụng tốt trên dữ liệu chưa thấy trước.

**Sau khi xử lý overfitting:**

- **Đường cong học tập của  $R^2$ :**

- **$R^2$  của tập huấn luyện giảm nhẹ** nhưng vẫn đủ cao để mô hình đạt hiệu suất tốt mà không ghi nhớ quá nhiều chi tiết không cần thiết. Điều này cho thấy mô hình đã giảm sự tập trung vào tối ưu hóa dữ liệu huấn luyện, tránh học quá mức các đặc điểm dư thừa.
- **$R^2$  trên tập kiểm tra tăng lên** và khoảng cách giữa  $R^2$  của hai tập thu hẹp lại, cho thấy mô hình đã tổng quát tốt hơn và có khả năng áp dụng tốt trên dữ liệu mới.

- **Chỉ số hiệu suất:**

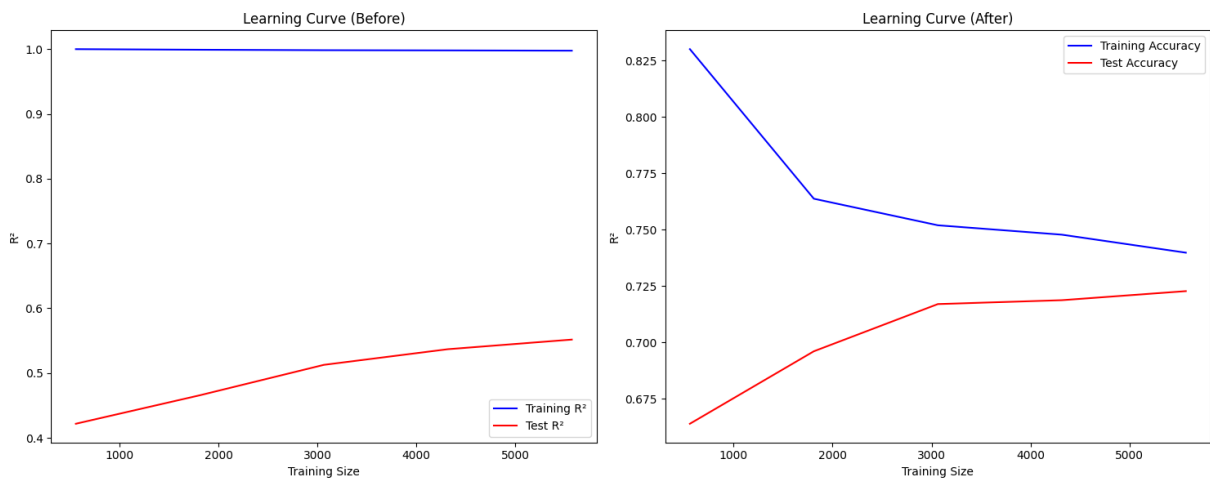
- **MAE và RMSE trên tập kiểm tra giảm**, chứng tỏ mô hình đã cải thiện độ chính xác trên dữ liệu mới. Giá trị của các chỉ số này có thể tăng nhẹ trên tập huấn luyện, điều này là bình thường khi giảm overfitting.
- **$R^2$  của tập kiểm tra tăng lên**, cho thấy mô hình đã cải thiện khả năng giải thích biến động của dữ liệu kiểm tra, phù hợp hơn với mục tiêu tổng quát hóa.

### 2.2.2 Phân loại

#### Decision Tree Classifier

Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

```
class_weight=None,criterion='gini',max_depth=None,
max_features=None,max_leaf_nodes=30,min_impurity_decrease=0.0,
min_samples_leaf=1 min_samples_split=2 splitter='best'
```



Hình 2-8 Learning Curve trước và sau điều chỉnh của Decision Tree Classifier

- **Trước khi xử lý overfitting:**

**R² của tập huấn luyện** rất cao (gần 1), cho thấy mô hình hoạt động tốt trên dữ liệu huấn luyện nhưng có thể đã ghi nhớ dữ liệu thay vì học các đặc điểm tổng quát. Đây là dấu hiệu rõ ràng của overfitting.

**R² của tập kiểm tra** thấp hơn nhiều (dao động từ 0.4 đến 0.55), cho thấy mô hình không dự đoán chính xác trên dữ liệu mới. Khoảng cách lớn giữa R² của tập huấn luyện và tập kiểm tra thể hiện vấn đề nghiêm trọng về overfitting.

- **Sau khi xử lý overfitting:**

**R² của tập huấn luyện** giảm từ mức gần 1 xuống khoảng 0.75-0.85, cho thấy mô hình không còn quá tập trung vào dữ liệu huấn luyện và đã bắt đầu tổng quát hóa.

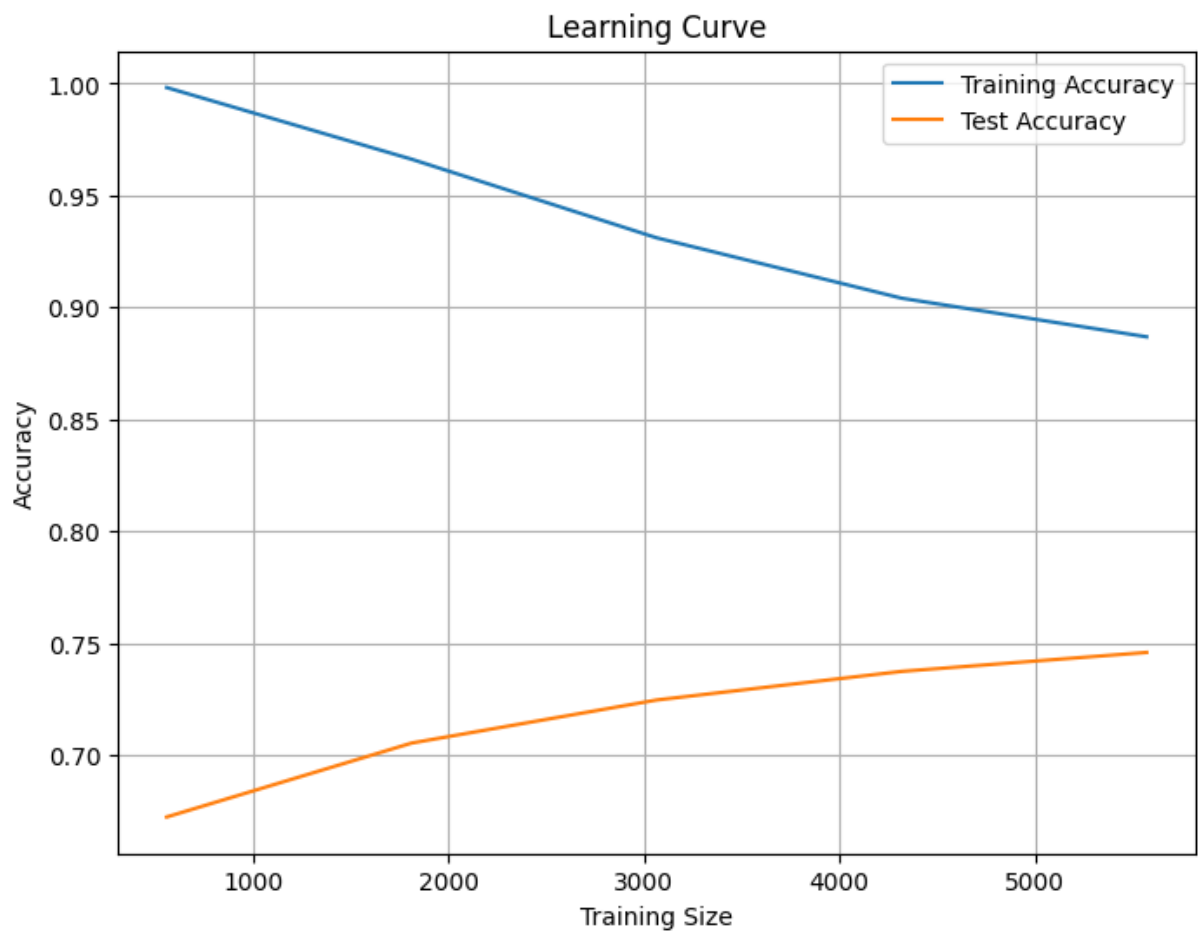


**$R^2$  của tập kiểm tra** tăng rõ rệt (lên khoảng 0.65-0.72), đồng thời khoảng cách giữa hai chỉ số  $R^2$  (train và test) giảm đáng kể. Điều này thể hiện mô hình đã cải thiện khả năng dự đoán và hoạt động hiệu quả hơn trên dữ liệu mới.

### 2.2.3 Sử dụng model mới cho phân loại

#### Dùng thêm XGBClassifier với bộ tham số

```
n_estimators= 200,  
max_depth= 6,  
learning_rate= 0.1,  
subsample= 0.8,  
colsample_bytree= 0.8,  
gamma= 0
```



Hình 2-9 Learning Curve trước và sau điều chỉnh của XGBClassifier

Dựa vào đường cong ta thấy model có khả năng học tập cao và không bị overfit

	Model	Accuracy
0	Logistic Regression	0.688777
1	Decision Tree	0.695477
2	KNN	0.578894

0.7695142378559464

Hình 2-10 Accuracy

Dựa vào ta thấy XGBClassifier hoạt động tốt hơn 3 model còn lại.

## CHƯƠNG 3 – CÂU 3

### 3.1 Giới thiệu feature selection using correlation analysis

#### 3.1.1 Feature selection là gì?

**Feature selection (lựa chọn đặc trưng)** là quá trình chọn ra một tập con các đặc trưng (features) quan trọng nhất từ bộ dữ liệu ban đầu để sử dụng trong việc xây dựng mô hình. Mục tiêu của feature selection là loại bỏ những đặc trưng không cần thiết, dư thừa hoặc ít ảnh hưởng đến kết quả dự đoán, giúp:

- Giảm độ phức tạp của mô hình: Sử dụng ít đặc trưng hơn giúp mô hình đơn giản hơn, từ đó giảm thời gian huấn luyện và yêu cầu tài nguyên.
- Cải thiện hiệu suất mô hình: Việc loại bỏ các đặc trưng không liên quan hoặc nhiễu có thể giúp tăng độ chính xác và khả năng tổng quát của mô hình.
- Giảm nguy cơ overfitting: Giảm số lượng đặc trưng giúp tránh tình trạng mô hình học quá mức vào chi tiết không cần thiết, cải thiện khả năng áp dụng của mô hình trên dữ liệu mới.
- Dễ hiểu và dễ phân tích hơn: Mô hình sẽ trở nên trực quan hơn khi chỉ sử dụng các đặc trưng quan trọng, giúp người dùng hiểu rõ hơn mối quan hệ giữa các biến đầu vào và đầu ra.

**Các phương pháp phổ biến trong feature selection gồm có:**

- Filter methods: Dựa vào các thước đo thống kê để chọn đặc trưng như độ tương quan, ANOVA, hay thống kê khi bình phương.
- Wrapper methods: Sử dụng các thuật toán để kiểm tra và chọn đặc trưng, như phương pháp forward selection, backward elimination, hay recursive feature elimination.

- **Embedded methods:** Các thuật toán có sẵn cơ chế chọn đặc trưng, như cây quyết định (Decision Tree), Lasso Regression, hay thuật toán máy học có regularization.

**Feature selection** là bước quan trọng giúp tối ưu hóa hiệu suất và độ chính xác của mô hình.

### 3.1.2 *Feature selection using correlation analysis là gì?*

**Feature selection using correlation analysis** (Lựa chọn đặc trưng thông qua phân tích tương quan) là một phương pháp lựa chọn đặc trưng trong đó các đặc trưng (features) được chọn hoặc loại bỏ dựa trên mức độ tương quan giữa chúng. Mục tiêu là loại bỏ những đặc trưng có mối quan hệ quá mạnh với nhau, vì chúng có thể gây ra hiện tượng đa cộng tuyến (multicollinearity) trong mô hình, làm giảm độ chính xác và hiệu quả dự đoán.

#### **Các bước thực hiện Feature Selection bằng phân tích tương quan:**

**Tính toán ma trận tương quan:** Đầu tiên, tính toán ma trận tương quan giữa các đặc trưng trong bộ dữ liệu. Ma trận này thể hiện mối quan hệ tuyến tính giữa từng cặp đặc trưng. Mỗi giá trị trong ma trận tương quan có thể dao động từ -1 đến 1:

**Tương quan gần 1 hoặc -1:** Có mối quan hệ mạnh (tương quan cao hoặc nghịch đảo).

**Tương quan gần 0:** Không có mối quan hệ tuyến tính rõ ràng.

#### **Loại bỏ đặc trưng có tương quan cao:**

Nếu hai đặc trưng có mức độ tương quan cao (thường là trên 0.8 hoặc 0.9), một trong số chúng có thể bị loại bỏ vì thông tin của chúng trùng lặp.

Việc loại bỏ đặc trưng giúp giảm sự dư thừa thông tin và làm cho mô hình trở nên đơn giản hơn.

**Giữ lại đặc trưng quan trọng:**

Những đặc trưng không có mối quan hệ tương quan cao với các đặc trưng khác nên được giữ lại vì chúng chứa thông tin độc lập.

Đặc trưng với sự tương quan thấp với các đặc trưng khác có thể có ảnh hưởng đáng kể đến mô hình.

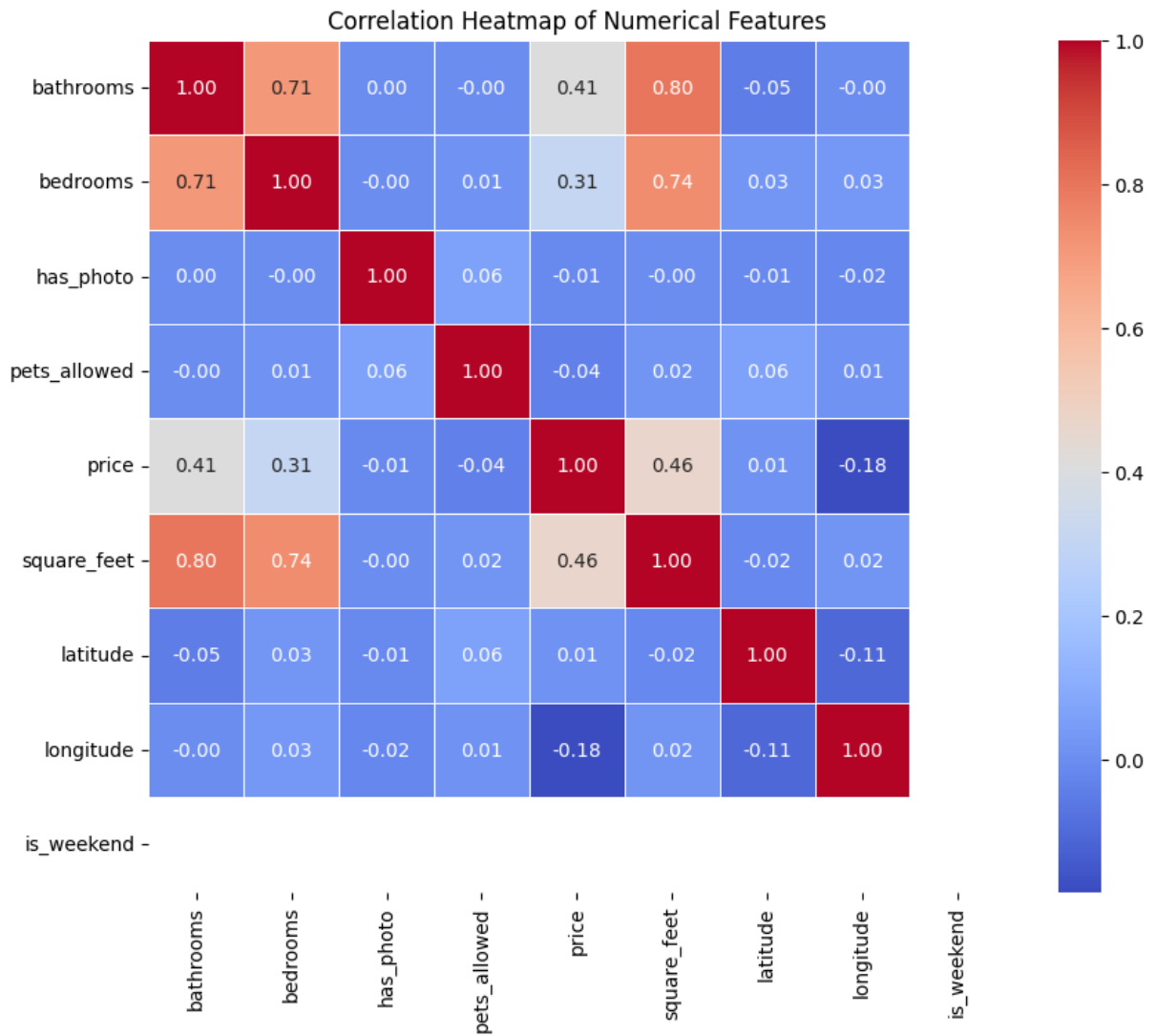
**Lợi ích của Feature Selection bằng phân tích tương quan:**

**Giảm đa cộng tuyến:** Khi các đặc trưng có mối quan hệ mạnh với nhau, chúng có thể gây ảnh hưởng xấu đến mô hình, đặc biệt là với các mô hình tuyến tính. Việc loại bỏ các đặc trưng có tương quan cao giúp giảm hiện tượng này.

**Tối ưu hóa hiệu suất mô hình:** Bằng cách loại bỏ các đặc trưng dư thừa, mô hình sẽ trở nên đơn giản hơn và có thể tổng quát tốt hơn trên dữ liệu mới.

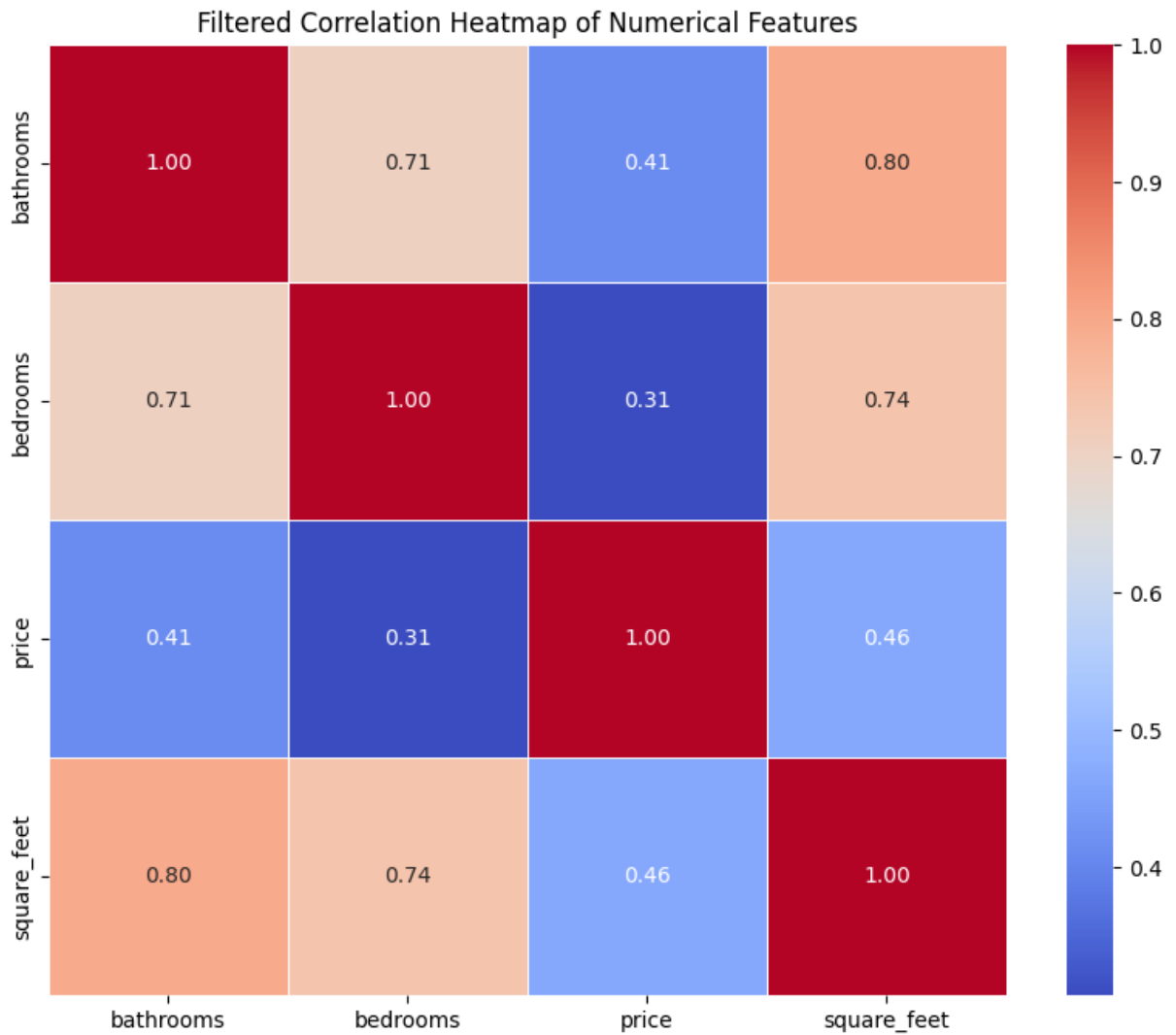
**Cải thiện khả năng giải thích mô hình:** Các đặc trưng có mối quan hệ độc lập với nhau giúp mô hình dễ hiểu và dễ giải thích hơn.

### 3.1.3 Áp dụng vào bài toán



Hình 3-1 Tương quan các thuộc tính trong dataset

Sau đó áp dụng xóa các thuộc tính có ngưỡng dưới 0.1 và trên 0.9



Hình 3-2 Tương quan của các thuộc tính lựa chọn

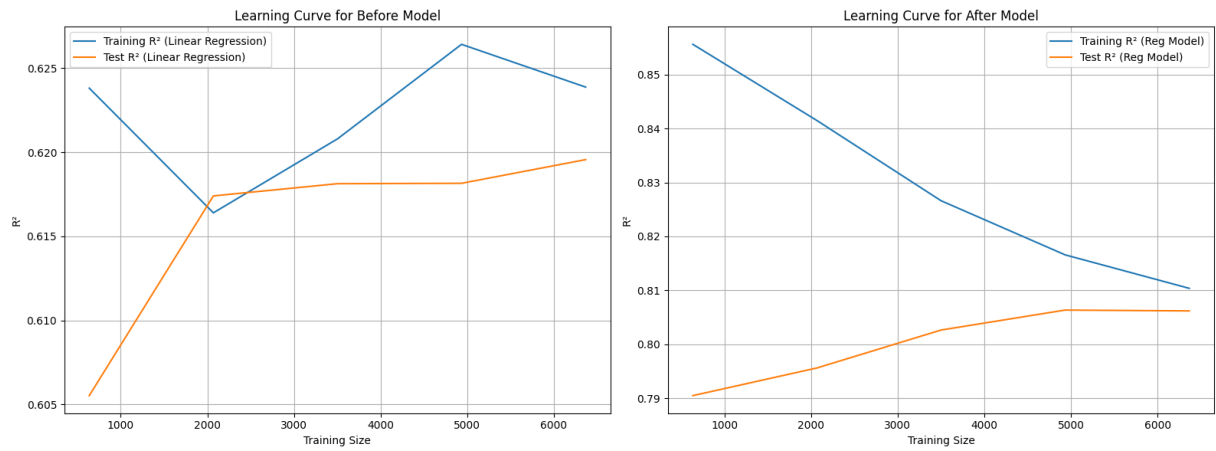
Lựa chọn các thuộc tính **bathrooms**, **bedrooms**, **price** để thực hiện hồi quy

	Model	$R^2$	MAE
0	Trước	0.617101	0.084822
1	Sau	0.735359	0.056357

Hình 3-3  $R^2$  và MAE trước và sau chọn thuộc tính



Sau khi áp dụng các thuộc tính được lựa chọn ta thấy các chỉ tăng lên rất nhiều, cho thấy model được cải thiện rất nhiều.



Hình 3-4 Learning Curve trước và sau điều chỉnh

Learning Curve cho thấy model sau khi lựa chọn đặc trưng có xu hướng học dữ liệu và tổng quan hóa tốt hơn model trước.

=> Từ đó kết luận được sau khi lựa chọn các đặc trưng nó hoạt động tốt hơn.

## **TÀI LIỆU THAM KHẢO**

### **Tiếng Việt**

1. Phan Minh Hoàng (2018), "Giới thiệu về học máy (Machine Learning) và ứng dụng", Tạp chí Khoa học Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.
2. Nguyễn Duy Hưng (2019), "Tổng quan về các phương pháp học sâu (Deep Learning)", Nhà xuất bản Giáo dục Việt Nam.
3. Vũ Đức Anh (2020), "Học máy trong xử lý dữ liệu lớn và ứng dụng", Luận văn Thạc sĩ Khoa học máy tính, Đại học Bách Khoa Hà Nội.

### **Tiếng Anh**

4. Mitchell, T. M. (1997), "Machine Learning", McGraw-Hill.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016), "Deep Learning", MIT Press.
6. Alpaydin, E. (2020), "Introduction to Machine Learning", MIT Press.
7. Bishop, C. M. (2006), "Pattern Recognition and Machine Learning", Springer.
8. Russell, S., & Norvig, P. (2016), "Artificial Intelligence: A Modern Approach", Pearson

## TỰ ĐÁNH GIÁ

	Phân công	Mức độ hoàn thành
Chung Thái Kiệt	1,2,3	100%
Huỳnh Thanh Bảo Ngọc	1,2,3	100%
Lê Hân	1,2,3	100%